Smooth minimization of nonsmooth functions with parallel coordinate descent methods

Smooth Minimization of Nonsmooth Functions with Parallel Coordinate Descent Methods

Olivier Fercoq * Peter Richtárik †

September 22, 2013

Abstract

We study the performance of a family of randomized parallel coordinate descent methods for minimizing the sum of a nonsmooth and separable convex functions. The problem class includes as a special case L1-regularized L1 regression and the minimization of the exponential loss ("AdaBoost problem"). We assume the input data defining the loss function is contained in a sparse $m \times n$ matrix A with at most ω nonzeros in each row. Our methods need $O(n\beta/\tau)$ iterations to find an approximate solution with high probability, where τ is the number of processors and $\beta = 1 + (\omega - 1)(\tau - 1)/(n - 1)$ for the fastest variant. The notation hides dependence on quantities such as the required accuracy and confidence levels and the distance of the starting iterate from an optimal point. Since β/τ is a decreasing function of τ, the method needs fewer iterations when more processors are used. Certain variants of our algorithms perform on average only $O(\mathrm{nnz}(A)/n)$ arithmetic operations during a single iteration per processor and, because β decreases when ω does, fewer iterations are needed for sparser problems.

1 Introduction

It is increasingly common that practitioners in machine learning, optimization, biology, engineering and various industries need to solve optimization problems with number of variables/coordinates so huge that classical algorithms, which for historical reasons almost invariably focus on obtaining solutions of high accuracy, are not efficient enough, or are outright unable to perform even a single iteration. Indeed, in the *big data optimization* setting, where the number N of variables is huge, inversion of matrices is not possible, and even operations such as matrix vector multiplications are too expensive. Instead, attention is shifting towards simple methods, with cheap iterations, low memory requirements and good parallelization and scalability properties.

If the accuracy requirements are moderate and the problem has only simple constraints (such as box constraints), methods with these properties do exist: *parallel coordinate descent methods* [2, 23, 26, 33] emerged as a very promising class of algorithms in this domain.

*School of Mathematics, The University of Edinburgh, United Kingdom (e-mail: olivier.fercoq@ed.ac.uk)

†School of Mathematics, The University of Edinburgh, United Kingdom (e-mail: peter.richtarik@ed.ac.uk)

The work of both authors was supported by the EPSRC grant EP/I017127/1 (Mathematics for Vast Digital Resources). The work of P.R. was also supported by the Centre for Numerical Algorithms and Intelligent Software (funded by EPSRC grant EP/G036136/1 and the Scottish Funding Council).

1.1 Parallel coordinate descent methods

In a recent paper [26], Richtárik and Takáč proposed and studied the complexity of a *parallel coordinate descent method (PCDM)* applied to the convex composite[1] optimization problem

$$\min_{x\in\mathbb{R}^N} \phi(x) + \Psi(x), \tag{1}$$

where $\phi : \mathbb{R}^N \to \mathbb{R}$ is an *arbitrary differentiable* convex function and $\Psi : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is a simple (block) *separable* convex regularizer, such as $\lambda\|x\|_1$. The N variables/coordinates of x are assumed to be partitioned into n blocks, $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ and PCDM at each iteration computes and applies updates to a randomly chosen subset $\hat{S} \subseteq [n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ of blocks (a "sampling") of the decision vector, *in parallel*. Formally, $\hat{S}$ is a random set-valued mapping with values in $2^{[n]}$.

PCDM encodes a family of algorithms where each variant is characterized by the probability law governing $\hat{S}$. The sets generated throughout the iterations are assumed to be independent and identically distributed. In this paper we focus on *uniform samplings*, which are characterized by the requirement that $\mathbf{P}(i \in \hat{S}) = \mathbf{P}(j \in \hat{S})$ for all $i, j \in [n]$. It is easy to see that for a uniform sampling one necessarily has[2]

$$\mathbf{P}(i \in \hat{S}) = \frac{\mathbf{E}[|\hat{S}|]}{n}. \tag{2}$$

In particular, we will focus on two special classes of uniform samplings: i) those for which $\mathbf{P}(|\hat{S}| = \tau) = 1$ (τ-*uniform samplings*), and ii) τ-uniform saplings with the additional property that all subsets of cardinality τ are chosen equally likely (τ-*nice samplings*). We will also say that a sampling is *proper* if $\mathbf{P}(|\hat{S}| \geq 1) > 0$.

It is clearly important to understand whether choosing $\tau > 1$, as opposed to $\tau = 1$, leads to acceleration in terms of an improved complexity bound. Richtárik and Takáč [26, Section 6] established *generic iteration complexity results* for PCDM applied to (1)—we describe them in some detail in Section 1.3. Let us only mention now that these results are generic in the sense that they hold under the blanket assumption that a certain inequality involving ϕ and $\hat{S}$ holds, so that if one is able to derive this inequality for a certain class of smooth convex functions ϕ, complexity results are readily available. The inequality (called Expected Separable Overapproximation, or ESO) is

$$\mathbf{E}\left[\phi(x + h_{[\hat{S}]})\right] \leq \phi(x) + \frac{\mathbf{E}[|\hat{S}|]}{n}\left(\langle\nabla\phi(x), h\rangle + \frac{\beta}{2}\sum_{i=1}^{n} w_i\langle B_i h^{(i)}, h^{(i)}\rangle\right), \qquad x, h \in \mathbb{R}^N, \tag{3}$$

where B_i are positive definite matrices (these can be chosen based on the structure of ϕ, or simply taken to be identities), $\beta > 0$, $w = (w_1, \ldots, w_n)$ is a vector of positive weights, and $h_{[\hat{S}]}$ denotes the random vector in $\mathbb{R}^N$ obtained from h by zeroing out all its blocks that do not belong to $\hat{S}$. That is, $h_{[S]}$ is the vector in $\mathbb{R}^N$ for which $h^{(i)}_{[S]} = h^{(i)}$ if $i \in S$ and $h^{(i)}_{[S]} = 0$, otherwise. When (3) holds, we say that ϕ admits a (β, w)-ESO with respect to $\hat{S}$. For simplicity, we may sometimes write $(\phi, \hat{S}) \sim \text{ESO}(\beta, w)$.

Let us now give the intuition behind the ESO inequality (3). Assuming the current iterate is x, PCDM changes $x^{(i)}$ to $x^{(i)} + h^{(i)}(x)$ for $i \in \hat{S}$, where $h(x)$ is the minimizer of the right hand side of (3). By doing so, we benefit from the following:

[1] Gradient methods for problems of this form were studied by Nesterov [21].

[2] This and other identities for block samplings were derived in [26, Section 3].

(i) Since the overapproximation is a convex quadratic in h, *it easy to compute* $h(x)$.

(ii) Since the overapproximation is block separable, one *can compute the updates* $h^{(i)}(x)$ *in parallel* for all $i \in \{1, 2, \ldots, n\}$.

(iii) For the same reason, one *can compute the updates or* $i \in S_k$ *only*, where S_k is the sample set drawn at iteration k following the law describing $\hat{S}$.

The algorithmic strategy of PCDM is to move to a new point in such a way that the expected value of the loss function evaluated at this new point is as small as possible. The method effectively decomposes the N-dimensional problem into n smaller convex quadratic problems, attending to a random subset of τ of them at each iteration, in parallel. A single iteration of PCDM can be compactly written as

$$x \leftarrow x + (h(x))_{[\hat{S}]}, \tag{4}$$

where $h(x) = (h^{(1)}(x), \ldots, h^{(n)}(x))$ and

$$h^{(i)}(x) = \arg\min_h \left\{ \langle (\nabla \phi(x))^{(i)}, h^{(i)} \rangle + \frac{\beta w_i}{2} \langle B_i h^{(i)}, h^{(i)} \rangle \right\} \overset{(3)}{=} -\frac{1}{\beta w_i} B_i^{-1} (\nabla \phi(x))^{(i)}. \tag{5}$$

From the update formula (5) we can see that $\frac{1}{\beta}$ can be interpreted as a stepsize. We would hence wish to choose small β, but not too small so that the method does not diverge. The issue of the computation of a good (small) parameter β is very intricate for several reasons, and is at the heart of the design of a randomized parallel coordinate descent method. Much of the theory developed in this paper is aimed at identifying a class of nonsmooth composite problems which, when smoothed, admit ESO with a small and easily computable value of β. In the following text we give some insight into why this issue is difficult, still in the simplified smooth setting.

1.2 Spurious ways of computing β

Recall that the parameters β and w giving rise to an ESO need to be *explicitly calculated* before the method is run as they are needed in the computation of the update steps. We will now describe the issues associated with finding suitable β, for simplicity assuming that w has been chosen/computed.

1. Let us start with a first approach to computing β. If the gradient of ϕ is Lipschitz with respect to the separable norm

$$\|x\|_w^2 \overset{\text{def}}{=} \sum_{i=1}^n w_i \langle B_i x^{(i)}, x^{(i)} \rangle,$$

with *known* Lipschitz constant L, then for all $x, h \in \mathbb{R}^N$ we have $\phi(x + h') \leq \phi(x) + \langle \nabla \phi(x), h' \rangle + \frac{L}{2} \|h'\|_w^2$. Now, if for fixed $h \in \mathbb{R}^N$ we substitute $h' = h_{[\hat{S}]}$ into this inequality, and take expectations utilizing the identities [26]

$$\mathbf{E}\left[\langle x, h_{[\hat{S}]} \rangle\right] = \frac{\mathbf{E}[|\hat{S}|]}{n} \langle x, h \rangle, \qquad \mathbf{E}\left[\|h_{[\hat{S}]}\|_w^2\right] = \frac{\mathbf{E}[|\hat{S}|]}{n} \|h\|_w^2, \tag{6}$$

we obtain $(\phi, \hat{S}) \sim \text{ESO}(\beta, w)$ for $\beta = L$. It turns out that this way of obtaining β is far from satisfactory, for several reasons.

(a) First, it is very difficult to compute L in the big data setting PCDMs are designed for. In the case of L2 regression, for instance, L will be equal to the largest eigenvalue of a certain $N \times N$ matrix. For huge N, this is a formidable task, and may actually be harder than the problem we are trying to solve.

(b) We show in Section 4.1 that taking $\beta = \frac{n}{\tau}c$, where c is a bound on the Lipschitz constants (with respect to the norm $\|\cdot\|_w$, at $h = 0$, uniform in x) of the gradients of the functions $h \to \mathbf{E}[\phi(x + h_{[\hat{S}]})]$ precisely characterizes (3), and leads to smaller (=better) values β. Surprisingly, this β can be $O(\sqrt{n})$ *times* smaller than L. As we shall see, this directly translates into iteration complexity speedup by the factor of $O(\sqrt{n})$.

2. It is often easy to obtain good β in the case $\tau = 1$. Indeed, it follows from [19, 24] that any smooth convex function ϕ will satisfy (3) with $\beta = 1$ and $w_i = L_i$, where L_i is the block Lipschitz constant of the gradient of ϕ with respect to the norm $\langle B_i \cdot, \cdot \rangle^{1/2}$, associated with block i. If the size of block i is N_i, then the computation of L_i will typically amount to the finding a maximal eigenvalue of an $N_i \times N_i$ matrix. If the block sizes N_i are sufficiently small, it is much simpler to compute n of these quantities than to compute L. Now, can we use a similar technique to obtain β in the $\tau > 1$ case? A naive idea would be to keep β unchanged ($\beta = 1$). In view of (5), this means that one would simply compute the updates $h^{(i)}(x)$ in the same way as in the $\tau = 1$ case, and apply them all. However, this strategy is doomed to fail: the method may end up oscillating between sub-optimal points (a simple 2 dimensional example was described in [33]). This issue arises since the algorithm overshoots: while the individual updates are safe for $\tau = 1$, it is not clear why adding them all up for arbitrary τ should decrease the function value.

3. A natural remedy to the problem described in §2 is to decrease the stepsize, i.e., to increase β as τ increases. In fact, it can be inferred from [26] that $\beta(\tau) = \tau$ always works: it satisfies the ESO inequality and the method converges. This makes intuitive sense since the actual step in the $\tau > 1$ case is obtained as the *average* of the block updates which are safe in the $\tau = 1$ case. By Jensen's inequality, this must decrease the objective function since the randomized serial method does (below we assume for notational simplicity that all blocks are of size one, e_i are the unit coordinate vectors):

$$\phi(x_+) = \phi\left(x - \sum_{i \in \hat{S}} \tfrac{1}{\tau L_i} (\nabla \phi(x))^{(i)} e_i\right) \leq \tfrac{1}{\tau} \sum_{i \in \hat{S}} \phi\left(x - \tfrac{1}{L_i} (\nabla \phi(x))^{(i)} e_i\right).$$

However, this approach compensates the increase of computational power (τ) by the same decrease in stepsize, which means that the parallel method ($\tau > 1$) might in the worst case require the same number of iterations as the serial one ($\tau = 1$).

4. The issues described in §2 and §3 lead us to the following question: Is it possible to *safely* and *quickly* choose/compute a value of β in the $\tau = 1$ case which is larger than 1 but smaller than τ? If this was possible, we could expect the parallel method to be much better than its serial counterpart. An affirmative answer to this question for the class of smooth convex partially separable functions ϕ was given in [26].

To summarize, the issue of selecting β in the parallel setting is very intricate, and of utmost significance for the algorithm. In the next two subsections we now give more insight into this issue and in doing so progress into discussing our contributions.

1.3 Generic complexity results and partial separability

The generic complexity results mentioned earlier, established in [26] for PCDM, have the form[3]

$$k \geq \left(\frac{\beta}{\tau}\right) \times n \times c \qquad \Rightarrow \qquad \mathbf{P}\left(\phi(x_k) - \min_x \phi(x) \leq \epsilon\right) \geq 1 - \rho,$$

where c is a constant independent of τ, and depending on the error tolerance ϵ, confidence tolerance ρ, initial iterate x_0, optimal point x^* and w. Moreover, c does not hide any large constants.

Keeping τ fixed, from (5) we see that larger values of β lead to smaller stepsizes. We commented earlier, appealing to intuition, that this translates into worse complexity. This is now affirmed and quantified by the above generic complexity result. Note, however, that this generic result *does not provide any concrete information about parallelization speedup* because it does not say anything about the dependence of β on τ. Clearly, parallelization speedup occurs when the function

$$T(\tau) = \frac{\beta(\tau)}{\tau}$$

is decreasing. The behavior of this function is important for big data problems which can only be solved by decomposition methods, such as PCDM, on modern HPC architectures.

Besides proving generic complexity bounds for PCDM, as outlined above, Richtárik and Takáč [26] identified a class of *smooth convex* functions ϕ for which β can be explicitly computed as a function of τ in closed form, and for which indeed $T(\tau)$ is decreasing: *partially separable* functions. A convex function $\phi : \mathbb{R}^N \to \mathbb{R}$ is partially separable of degree ω if it can be written as a sum of differentiable[4] convex functions, each of which depends on at most ω of the n blocks of x. If $\hat{S}$ is a τ-uniform sampling, then $\beta = \beta' = \min\{\omega, \tau\}$. If $\hat{S}$ is a τ-nice sampling, then $\beta = \beta'' = 1 + \frac{(\omega-1)(\tau-1)}{n-1}$. Note that $\beta'' \leq \beta'$ and that β' can be arbitrarily larger than β''. Indeed, the worst case situation (in terms of the ratio $\frac{\beta'}{\beta''}$) for any fixed n is $\omega = \tau = \sqrt{n}$, in which case

$$\frac{\beta'}{\beta''} = \frac{1 + \sqrt{n}}{2}.$$

This means that PCDM implemented with a τ-nice sampling (using β'') can be arbitrarily faster than PCDM implemented with the more general τ-uniform sampling (using β'). This simple example illustrates the huge impact the choice of the sampling $\hat{S}$ has, other things equal. As we shall show in this paper, this phenomenon is directly related to the issue we discussed in Section 1.2: L can be $O(\sqrt{n})$ times larger than a good β.

[3] This holds provided w does not change with τ; which is the case in this paper and in the smooth partially separable setting considered in [26, Section 6]. Also, for simplicity we cast the results here in the case $\Psi \equiv 0$, but they hold in the composite case as well.

[4] It is not assumed that the summands have Lipschitz gradient.

1.4 Brief literature review

Serial randomized methods. Leventhal and Lewis [10] studied the complexity of randomized coordinate descent methods for the minimization of convex quadratics and proved that the method converges linearly even in the non-strongly convex case. Linear convergence for smooth strongly convex functions was proved by Nesterov [19] and for general regularized problems by Richtárik and Takáč [24]. Complexity results for smooth problems with special regularizes (box constraints, L1 norm) were obtained by Shalev-Shwarz and Tewari [30] and Nesterov [19]. Nesterov was the first to analyze the block setting, and proposed using different Lipschitz constants for different blocks, which has a big impact on the efficiency of the method since these constants capture important second order information [19]. Also, he was the first to analyze an accelerated coordinate descent method. Richtárik and Takáč [25, 24] improved, generalized and simplified previous results and extended the analysis to the composite case. They also gave the first analysis of a coordinate descent method using arbitrary probabilities. Lu and Xiao [11] recently studied the work developed in [19] and [26] and obtained further improvements. Coordinate descent methods were recently extended to deal with coupled constraints by Necoara et al [16] and extended to the composite setting by Necoara and Patrascu [17]. When the function is not smooth neither composite, it is still possible to define coordinate descent methods with subgradients. An algorithm based on the averaging of past subgradient coordinates is presented in [34] and a successful subgradient-based coordinate descent method for problems with sparse subgradients is proposed by Nesterov [20]. Tappenden et al [36] analyzed an inexact randomized coordinate descent method in which proximal subproblems at each iteration are solved only approximately. Dang and Lan [4] studied complexity of stochastic block mirror descent methods for nonsmooth and stochastic optimization and an accelerated method was studies by Shalev-Shwarz and Zhang [31]. Lacoste-Julien et al [9] were the first to develop a block-coordinate Frank-Wolfe method. The generalized power method of Journée et al [8] designed for sparse PCA can be seen as a nonconvex block coordinate ascent method with two blocks [27].

Parallel methods. One of the first complexity results for a parallel coordinate descent method was obtained by Ruszczyński [28] and is known as the diagonal quadratic approximation method (DQAM). DQAM updates all blocks at each iteration, and hence is not randomized. The method was designed for solving a convex composite problem with quadratic smooth part and arbitrary separable nonsmooth part and was motivated by the need to solve separable linearly constrained problems arising in stochastic programming. As described in previous sections, a family of randomized parallel block coordinate descent methods (PCDM) for convex composite problems was analyzed by Richtárik and Takáč [26]. Tappenden et al [35] recently contrasted the DQA method [28] with PCDM [26], improved the complexity result [26] in the strongly convex case and showed that for PCDM it is optimal choose τ to be equal to the number of processors. Utilizing the ESO machinery [26] and the primal-dual technique developed by Shalev-Shwarz and Zhang [32], Takáč et al [33] developed and analyzed a parallel (mini-batch) stochastic subgradient descent method (applied to the primal problem of training support vector machines with the hinge loss) and a parallel stochastic dual coordinate ascent method (applied to the dual box-constrained concave maximization problem). The analysis naturally extends to the general setting of Shalev-Shwarz and Zhang [32]. A parallel Newton coordinate descent method was proposed in [1]. Parallel methods for L1 regularized problems with an application to truss topology design were proposed by Richtárik and Takáč [23]. They give the first analysis of a greedy serial coordinate descent method for L1 regularized problems. An early analysis of a PCDM for L1 regularized problems was performed by Bradley et al [2]. Other recent parallel methods include [15, 13].

1.5 Contents

In Section 2 we describe the problems we study, the algorithm (smoothed parallel coordinate descent method), review Nesterov's smoothing technique and enumerate our contributions. In Section 3 we compute Lipschitz constants of the gradient smooth approximations of Nesterov separable functions associated with subspaces spanned by arbitrary subset of blocks, and in Section 4 we derive ESO inequalities. Complexity results are derived in Section 5 and finally, in Section 6 we describe three applications and preliminary numerical experiments.

2 Smoothed Parallel Coordinate Descent Method

In this section we describe the problems we study, the algorithm and list our contributions.

2.1 Nonsmooth and smoothed composite problems

In this paper we study the iteration complexity of PCDMs applied to two classes of convex composite optimization problems:

$$\text{minimize} \quad F(x) \stackrel{\text{def}}{=} f(x) + \Psi(x) \quad \text{subject to} \quad x \in \mathbb{R}^N, \tag{7}$$

and

$$\text{minimize} \quad F_\mu(x) \stackrel{\text{def}}{=} f_\mu(x) + \Psi(x) \quad \text{subject to} \quad x \in \mathbb{R}^N. \tag{8}$$

We assume (7) has an optimal solution (x^*) and consider the following setup:

1. **(Structure of f)** First, we assume that f is of the form

$$f(x) \stackrel{\text{def}}{=} \max_{z \in Q} \{\langle Ax, z\rangle - g(z)\}, \tag{9}$$

where $Q \subseteq \mathbb{R}^m$ is a nonempty compact convex set, $A \in \mathbb{R}^{m\times N}$, $g : \mathbb{R}^m \to \mathbb{R}$ is convex and $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product (the sum of products of the coordinates of the vectors). Note that f is convex and in general *nonsmooth*.

2. **(Structure of f_μ)** Further, we assume that f_μ is of the form

$$f_\mu(x) \stackrel{\text{def}}{=} \max_{z \in Q} \{\langle Ax, z\rangle - g(z) - \mu d(z)\}, \tag{10}$$

where A, Q and g are as above, $\mu > 0$ and $d : \mathbb{R}^m \to \mathbb{R}$ is σ-strongly convex on Q with respect to the norm

$$\|z\|_v \stackrel{\text{def}}{=} \Big(\sum_{j=1}^m v_j^p |z_j|^p\Big)^{1/p}, \tag{11}$$

where $v_1, \ldots, v_m$ are positive scalars, $1 \le p \le 2$ and $z = (z_1, \ldots, z_m)^T \in \mathbb{R}^m$. We further assume that d is nonnegative on Q and that $d(z_0) = 0$ for some $z_0 \in Q$. It then follows that $d(z) \ge \frac{\sigma}{2}\|z - z_0\|_v^2$ for all $z \in Q$. That is, d is a prox function on Q. We further let $D \stackrel{\text{def}}{=} \max_{z\in Q} d(z)$.

For $p > 1$ let q be such that $\frac{1}{p} + \frac{1}{q} = 1$. Then the conjugate norm of $\|\cdot\|_v$ defined in (11) is given by

$$\|z\|_v^* \stackrel{\text{def}}{=} \max_{\|z'\|_v \le 1} \langle z', z\rangle = \begin{cases} \left(\sum_{j=1}^m v_j^{-q}|z_j|^q\right)^{1/q}, & 1 < p \le 2, \\ \max_{1\le j\le m} v_j^{-1}|z_j|, & p = 1. \end{cases} \tag{12}$$

It is well known that f_μ is a *smooth* convex function; i.e., it is differentiable and its gradient is Lipschitz.

Remark: As shown by Nesterov in his seminal work on smooth minimization of nonsmooth functions [18]—here summarized in Proposition 2—f_μ is a smooth approximation of f. In this paper, when solving (7), we apply PCDM to (8) for a specific choice of $\mu > 0$, and then argue, following now-standard reasoning from [18], that the solution is an approximate solution of the original problem. This will be made precise in Section 2.2. However, in some cases one is interested in minimizing a function of the form (8) directly, without the need to interpret f_μ as a smooth approximation of another function. For instance, as we shall see in Section 6.3, this is the case with the "AdaBoost problem". In summary, both problems (7) and (8) are of interest on their own, even though our approach to solving the first one is by transforming it to the second one.

3. **(Block structure)** Let $A = [A_1, A_2, \ldots, A_n]$ be decomposed into nonzero column submatrices, where $A_i \in \mathbb{R}^{m\times N_i}$, $N_i \ge 1$ and $\sum_{i=1}^n N_i = N$, and $U = [U_1, U_2, \ldots, U_n]$ be a decomposition of the $N \times N$ identity matrix U into submatrices $U_i \in \mathbb{R}^{N\times N_i}$. Note that

$$A_i = AU_i. \tag{13}$$

It will be useful to note that

$$U_i^T U_j = \begin{cases} N_i \times N_i \text{ identity matrix}, & i = j, \\ N_i \times N_j \text{ zero matrix}, & \text{otherwise.} \end{cases} \tag{14}$$

For $x \in \mathbb{R}^N$, let $x^{(i)}$ be the block of variables corresponding to the columns of A captured by A_i, that is, $x^{(i)} = U_i^T x \in \mathbb{R}^{N_i}$, $i = 1, 2, \ldots, n$. Clearly, any vector $x \in \mathbb{R}^N$ can be written uniquely as $x = \sum_{i=1}^n U_i x^{(i)}$. We will often refer to the vector $x^{(i)}$ as the *i-th block* of x. We can now formalize the notation used in the introduction (e.g., in (4)): for $h \in \mathbb{R}^N$ and $\emptyset \ne S \subseteq [n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ it will be convenient to write

$$h_{[S]} \stackrel{\text{def}}{=} \sum_{i\in S} U_i h^{(i)}. \tag{15}$$

Finally, with each block i we associate a positive definite matrix $B_i \in \mathbb{R}^{N_i\times N_i}$ and scalar $w_i > 0$, and equip $\mathbb{R}^N$ with a pair of conjugate norms:

$$\|x\|_w^2 \stackrel{\text{def}}{=} \sum_{i=1}^n w_i \langle B_i x^{(i)}, x^{(i)}\rangle, \qquad (\|y\|_w^*)^2 \stackrel{\text{def}}{=} \max_{\|x\|_w \le 1} \langle y, x\rangle^2 = \sum_{i=1}^n w_i^{-1} \langle B_i^{-1} y^{(i)}, y^{(i)}\rangle. \tag{16}$$

Remark: For some problems, it is relevant to consider blocks of coordinates as opposed to individual coordinates. The novel aspects of this paper are *not* in the block setup however, which was already considered in [19, 26]. We still write the paper in the general block setting; for several reasons. First, it is often practical to work with blocks either due to the nature of the problem (e.g., group lasso), or due to numerical considerations

(it is often more efficient to process a "block" of coordinates at the same time). Moreover, some parts of the theory need to be treated differently in the block setting. The theory, however, does not get more complicated due to the introduction of blocks. A small notational overhead is a small price to pay for these benefits.

4. **(Sparsity of A)** For a vector $x \in \mathbb{R}^N$ let

$$\Omega(x) \stackrel{\text{def}}{=} \{i \ : \ U_i^T x \neq 0\} = \{i \ : \ x^{(i)} \neq 0\}. \tag{17}$$

Let A_{ji} be the j-th row of A_i. If $e_1, \ldots, e_m$ are the unit coordinate vectors in $\mathbb{R}^m$, then

$$A_{ji} \stackrel{\text{def}}{=} e_j^T A_i. \tag{18}$$

Using the above notation, the set of nonzero blocks of the j-th row of A can be expressed as

$$\Omega(A^T e_j) \stackrel{(17)}{=} \{i \ : \ U_i^T A^T e_j \neq 0\} \stackrel{(13)+(18)}{=} \{i \ : \ A_{ji} \neq 0\}. \tag{19}$$

The following concept is key to this paper.

Definition 1 (Nesterov separability[5])**.** We say that f (resp. f_μ) is *Nesterov (block) separable* of degree ω if it has the form (9) (resp. (10)) and

$$\max_{1 \leq j \leq m} |\Omega(A^T e_j)| \leq \omega. \tag{20}$$

Note that in the special case when all blocks are of cardinality 1 (i.e., $N_i = 1$ for all i), the above definition simply requires all rows of A to have at most ω nonzero entries.

5. **(Separability of Ψ)** We assume that

$$\Psi(x) = \sum_{i=1}^{n} \Psi_i(x^{(i)}),$$

where $\Psi_i : \mathbb{R}^{N_i} \to \mathbb{R} \cup \{+\infty\}$ are *simple* proper closed convex functions.

Remark: Note that we do not assume that the functions Ψ_i be smooth. In fact, the most interesting cases in terms of applications are nonsmooth functions such as, for instance, i) $\Psi_i(t) = \lambda|t|$ for some $\lambda > 0$ and all i (L1 regularized optimization), ii) $\Psi_i(t) = 0$ for $t \in [a_i, b_i]$, where $-\infty \leq a_i \leq b_i \leq +\infty$ are some constants, and $\Psi_i(t) = +\infty$ for $t \notin [a_i, b_i]$ (box constrained optimization).

We are now ready to state the method (Algorithm 1) we use for solving the smoothed composite problem (8). Note that for $\phi \equiv f_\mu$ and $\Psi \equiv 0$, Algorithm 1 coincides with the method (4)-(5) described in the introduction. The only conceptual difference here is that in the computation of the updates in Step 2 we need to augment the quadratic obtained from ESO with Ψ. Note that Step 3 can be compactly written as

$$x_{k+1} = x_k + (h_k)_{[S_k]}. \tag{21}$$

[5]We coined the term *Nesterov separability* in honor of Yu. Nesterov's seminal work on the smoothing technique [18], which is applicable to functions represented in the form (9). Nesterov did not study problems with row-sparse matrices A, as we do in this work, nor did he study parallel coordinate descent methods. However, he proposed the celebrated smoothing technique which we also employ in this paper.

Algorithm 1 Smoothed Parallel Coordinate Descent Method (SPCDM)

Input: initial iterate $x_0 \in \mathbb{R}^N$, $\beta > 0$ and $w = (w_1, \dots, w_n) > 0$

for $k \geq 0$ **do**

Step 1. Generate a random set of blocks $S_k \subseteq \{1, 2, \dots, n\}$

Step 2. In parallel for $i \in S_k$, compute

$$h_k^{(i)} = \arg\min_{t \in \mathbb{R}^{N_i}} \left\{ \langle (\nabla f_\mu(x_k))^{(i)}, t \rangle + \frac{\beta w_i}{2} \langle B_i t, t \rangle + \Psi_i(x_k^{(i)} + t) \right\}$$

Step 3. In parallel for $i \in S_k$, update $x_k^{(i)} \leftarrow x_k^{(i)} + h_k^{(i)}$ and set $x_{k+1} \leftarrow x_k$

end for

Let us remark that the scheme actually encodes an entire family of methods. For $\tau = 1$ we have a serial method (one block updated per iteration), for $\tau = n$ we have a fully parallel method (all blocks updated in each iteration), and there are many partially parallel methods in between, depending on the choice of τ. Likewise, there is flexibility in choosing the block structure. For instance, if we choose $N_i = 1$ for all i, we have a proximal coordinate descent method, for $N_i > 1$, we have a proximal block coordinate descent and for $n = 1$ we have a proximal gradient descent method.

2.2 Nesterov's smoothing technique

In the rest of the paper we will repeatedly make use of the now-classical smoothing technique of Nesterov [18]. We will *not* use this merely to approximate f by f_μ; the technique will be utilized in several proofs in other ways, too. In this section we collect the facts that we will need.

Let $\mathbb{E}_1$ and $\mathbb{E}_2$ be two finite dimensional linear normed spaces, and $\mathbb{E}_1^*$ and $\mathbb{E}_2^*$ be their duals (i.e., the spaces of bounded linear functionals). We equip $\mathbb{E}_1$ and $\mathbb{E}_2$ with norms $\|\cdot\|_1$ and $\|\cdot\|_2$, and the dual spaces $\mathbb{E}_1^*$, $\mathbb{E}_2^*$ with the dual (conjugate norms):

$$\|y\|_j^* \stackrel{\text{def}}{=} \max_{\|x\|_j \leq 1} \langle y, x \rangle, \qquad y \in \mathbb{E}_j^*, \qquad j = 1, 2,$$

where $\langle y, x \rangle$ denotes the action of the linear functional y on x. Let $\bar{A} : \mathbb{E}_1 \to \mathbb{E}_2^*$ be a linear operator, and let $\bar{A}^* : \mathbb{E}_2 \to \mathbb{E}_1^*$ be its adjoint:

$$\langle \bar{A}x, u \rangle = \langle x, \bar{A}^* u \rangle, \qquad x \in \mathbb{E}_1, \qquad u \in \mathbb{E}_2.$$

Let us equip $\bar{A}$ with a norm as follows:

$$\begin{aligned} \|\bar{A}\|_{1,2} &\stackrel{\text{def}}{=} \max_{x,u} \{ \langle Ax, u \rangle \ : \ x \in \mathbb{E}_1, \ \|x\|_1 = 1, \ u \in \mathbb{E}_2, \ \|u\|_2 = 1 \} \\ &= \max_x \{ \|\bar{A}x\|_2^* \ : \ x \in \mathbb{E}_1, \ \|x\|_1 = 1 \} = \max_u \{ \|\bar{A}^* u\|_1^* \ : \ u \in \mathbb{E}_2, \ \|u\|_2 = 1 \}. \end{aligned} \tag{22}$$

Consider now the function $\bar{f} : \mathbb{E}_1 \to \mathbb{R}$ given by

$$\bar{f}(x) = \max_{u \in \bar{Q}} \{ \langle \bar{A}x, u \rangle - \bar{g}(u) \},$$

where $\bar{Q} \subset \mathbb{E}_2$ is a compact convex set and $\bar{g} : \mathbb{E}_2 \to \mathbb{R}$ is convex. Clearly, $\bar{f}$ is convex and in general nonsmooth.

We now describe Nesterov's smoothing technique for approximating $\bar{f}$ by a convex function with Lipschitz gradient. The technique relies on the introduction of a prox-function $\bar{d} : \mathbb{E}_2 \to \mathbb{R}$. This function is continuous and strongly convex on $\bar{Q}$ with convexity parameter $\bar{\sigma}$. Let u_0 be the minimizer of $\bar{d}$ on $\bar{Q}$. Without loss of generality, we can assume that $\bar{d}(u_0) = 0$ so that for all $u \in \bar{Q}$, $\bar{d}(u) \geq \frac{\bar{\sigma}}{2}\|u - u_0\|_2^2$. We also write $\bar{D} \stackrel{\text{def}}{=} \max\{\bar{d}(u) \ : \ u \in \bar{Q}\}$. Nesterov's smooth approximation of $\bar{f}$ is defined for any $\mu > 0$ by

$$\bar{f}_\mu(x) \stackrel{\text{def}}{=} \max_{u \in \bar{Q}} \{\langle \bar{A}x, u\rangle - \bar{g}(u) - \mu \bar{d}(u)\}. \tag{23}$$

Proposition 2 (Nesterov [18])**.** *The function $\bar{f}_\mu$ is continuously differentiable on $\mathbb{E}_1$ and satisfies*

$$\bar{f}_\mu(x) \leq \bar{f}(x) \leq \bar{f}_\mu(x) + \mu \bar{D}. \tag{24}$$

Moreover, $\bar{f}_\mu$ is convex and its gradient $\nabla \bar{f}_\mu(x) = \bar{A}^* u^*$, where u^* is the unique maximizer in (23)*, is Lipschitz continuous with constant*

$$L_\mu = \frac{1}{\mu \bar{\sigma}} \|\bar{A}\|_{1,2}^2. \tag{25}$$

That is, for all $x, h \in \mathbb{E}_1$,

$$\bar{f}_\mu(x + h) \leq \bar{f}_\mu(x) + \langle \nabla \bar{f}_\mu(x), h\rangle + \frac{\|\bar{A}\|_{1,2}^2}{2\mu\bar{\sigma}} \|h\|_1^2. \tag{26}$$

The above result will be used in this paper in various ways:

1. As a direct consequence of (26) for $\mathbb{E}_1 = \mathbb{R}^N$ (primal basic space), $\mathbb{E}_2 = \mathbb{R}^m$ (dual basic space), $\|\cdot\|_1 = \|\cdot\|_w$, $\|\cdot\|_2 = \|\cdot\|_v$, $\bar{d} = d$, $\bar{\sigma} = \sigma$, $\bar{Q} = Q$, $\bar{g} = g$, $\bar{A} = A$ and $\bar{f} = f$, we obtain the following inequality:

$$f_\mu(x + h) \leq f_\mu(x) + \langle \nabla f_\mu(x), h\rangle + \frac{\|A\|_{w,v}^2}{2\mu\sigma} \|h\|_w^2. \tag{27}$$

2. A large part of this paper is devoted to various refinements (for a carefully chosen data-dependent w we "replace" $\|A\|_{w,v}^2$ by an easily computable and interpretable quantity depending on h and ω, which gets smaller as h gets sparser and ω decreases) and extensions (left-hand side is replaced by $\mathbf{E}[f_\mu(x + h_{[\hat{S}]})]$) of inequality (27). In particular, we give formulas for fast computation of subspace Lipschitz constants of ∇f_μ (Section 3) and derive ESO inequalities (Section 4)—which are essential for proving iteration complexity results for variants of the smoothed parallel coordinate descent method (Algorithm 1).

3. Besides the above application to smoothing f; we will utilize Proposition 2 also as a tool for computing Lipschitz constants of the gradient of two technical functions needed in proofs. In Section 3 we will use $\mathbb{E}_1 = \mathbb{R}^S$ ("primal update space" associated with a subset $S \subseteq [n]$), $\mathbb{E}_2 = \mathbb{R}^m$ and $\bar{A} = A^{(S)}$. In Section 4 we will use $\mathbb{E}_1 = \mathbb{R}^N$, $\mathbb{E}_2 = \mathbb{R}^{|\mathcal{P}|\times m}$ ("dual product space" associated with sampling $\hat{S}$) and $\bar{A} = \hat{A}$. These spaces and matrices will be defined in the above mentioned sections, where they are needed.

The following simple consequence of Proposition 2 will be useful in proving our complexity results.

Lemma 3. *Let x^* be an optimal solution of (7) (i.e., $x^* = \arg\min_x F(x)$) and x^*_μ be an optimal solution of (8) (i.e., $x^*_\mu = \arg\min_x F_\mu(x)$). Then for any $x \in \operatorname{dom}\Psi$ and $\mu > 0$,*

$$F_\mu(x) - F_\mu(x^*_\mu) - \mu D \quad \leq \quad F(x) - F(x^*) \quad \leq \quad F_\mu(x) - F_\mu(x^*_\mu) + \mu D. \tag{28}$$

Proof. From Proposition 2 (used with $\bar{A} = A$, $\bar{f} = f$, $\bar{Q} = Q$, $\bar{d} = d$, $\|\cdot\|_2 = \|\cdot\|_v$, $\bar{\sigma} = \sigma$, $\bar{D} = D$ and $\bar{f}_\mu = f_\mu$), we get $f_\mu(y) \leq f(y) \leq f_\mu(y) + \mu D$, and adding $\Psi(y)$ to all terms leads to $F_\mu(y) \leq F(y) \leq F_\mu(y) + \mu D$, for all $y \in \operatorname{dom}\Psi$. We only prove the second inequality, the first one can be shown analogously. From the last chain of inequalities and optimality of x^*_μ we get i) $F(x) \leq F_\mu(x) + \mu D$ and ii) $F_\mu(x^*_\mu) \leq F_\mu(x^*) \leq F(x^*)$. We only need to subtract (ii) from (i). □

2.3 Contributions

We now describe some of the main contributions of this work.

1. **First complexity results.** We give the first complexity results for solving problems (7) and (8) by a parallel coordinate descent method. In fact, to the best of our knowledge, we are not aware of any complexity results even in the $\Psi \equiv 0$ case. We obtain our results by combining the following: i) we show that f_μ—smooth approximation of f—admits ESO inequalities with respect to uniform samplings and compute "good" parameters β and w, ii) for problem (7) we utilize Nesterov's smoothing results (via Lemma (3)) to argue that an approximate solution of (8) is an approximate solution of (7), iii) we use the generic complexity bounds proved by Richtárik and Takáč [26].

2. **Nesterov separability.** We identify the degree of *Nesterov separability* as the important quantity driving parallelization speedup.

3. **ESO parameters.** We show that it is possible to compute ESO parameters β and w *easily*. This is of utmost importance for big data applications where the computation of the Lipschitz constant L of $\nabla\phi = \nabla f_\mu$ is prohibitively expensive (recall the discussion in Section 1.2). In particular, we suggest that in the case with all blocks being of size 1 ($N_i = 1$ and $B_i = 1$ for all i), the weights $w_i = w_i^*$, $i = 1, 2, \ldots, n$, be chosen as follows:

$$w_i^* = \begin{cases} \max_{1\leq j\leq m} v_j^{-2} A_{ji}^2, & p = 1, \\ \left(\sum_{j=1}^m v_j^{-q} |A_{ji}|^q\right)^{2/q}, & 1 < p < 2, \\ \sum_{j=1}^m v_j^{-2} A_{ji}^2, & p = 2. \end{cases} \tag{29}$$

   These weights can be computed in $O(\mathrm{nnz}(A))$ time. The general formula for w^* for arbitrary blocks and matrices B_i is given in (38).

   Moreover, we show (Theorems 13 and 15) that $(f_\mu, \hat{S}) \sim \mathrm{ESO}(\beta, w^*)$, where $\beta = \frac{\beta'}{\sigma\mu}$ and

$$\beta' = \begin{cases} \min\{\omega, \tau\}, & \text{if } \hat{S} \text{ is } \tau\text{-uniform}, \\ 1 + \frac{(\omega-1)(\tau-1)}{\max\{1,n-1\}}, & \text{if } \hat{S} \text{ is } \tau\text{-nice and } p = 2, \end{cases}$$

and ω is the degree of Nesterov separability. The formula for β' in the case of a τ-nice sampling $\hat{S}$ and $p = 1$ is more involved and is given in Theorem 15. This value is always larger than β' in the $p = 2$ case (recall that small β' is better), and increases with m. However, they are often very close in practice (see Figure 1).

Surprisingly, the formulas for β' in the two cases summarized above are identical to those obtained in [26] for smooth partially separable functions (recall the discussion in Section 1.3), although the classes of functions considered are *different*. The investigation of this phenomenon is an open question.

We also give formulas for β for arbitrary w, but these involve the computation of a complicated matrix norm (Theorem 11). The above formulas for β are *good* (in terms of the parallelization speedup they lead to), *easily computable* and *interpretable* bounds on this norm for $w = w^*$.

4. **Complexity.** Our complexity results are spelled out in detail in Theorems 16 and 17, and are summarized in the table below.

	strong convexity	convexity
Problem 7 [Thm 16]	$\frac{n}{\tau} \times \frac{\frac{\beta'}{\mu\sigma}+\sigma_\Psi}{\sigma_{f_\mu}+\sigma_\Psi}$	$\frac{n\beta'}{\tau} \times \frac{2Diam^2}{\mu\sigma\epsilon}$
Problem 8 [Thm 17]	$\frac{n}{\tau} \times \frac{\frac{2\beta' D}{\epsilon\sigma}+\sigma_\Psi}{\sigma_{f_\mu}+\sigma_\Psi}$	$\frac{n\beta'}{\tau} \times \frac{8DDiam^2}{\sigma\epsilon^2}$

The results are complete up to logarithmic factors and say that as long as SPCDM takes at least k iterations, where lower bounds for k are given in the table, then x_k is an ϵ-solution with probability at least $1 - \rho$. The confidence level parameter ρ can't be found in the table as it appears in a logarithmic term which we suppressed from the table. For the same reason, it is easy for SPCDM to achieve arbitrarily high confidence. More on the parameters: n is then umber of blocks, σ, μ and D are defined in §2 of Section 2.1. The remaining parameters will be defined precisely in Section 5: σ_ϕ denotes the strong convexity constant of ϕ with respect to the norm $\|\cdot\|_{w^*}$ (for $\phi = \Psi$ and $\phi = f_\mu$) and $Diam$ is the diameter of the level set of the loss function defined by the value of the loss function at the initial iterate x_0.

Observe that as τ increases, the number of iteration decreases. The actual rate of decrease is controlled by the value of β' (as this is the only quantity that may grow with τ). In the convex case, any value of β' smaller than τ leads to parallelization speedup. Indeed, as we discussed in §3 above, the values of β' are much smaller than τ, and decrease to 1 as ω approaches 1. Hence, the more separable the problem is, in terms of the degree of partial separability ω, the better. In the strongly convex case, the situation is even better.

5. **Cost of a single iteration.** The arithmetic cost of a single iteration of SPCDM is $c = c_1 + c_2 + c_3$, where c_1 is the cost of computing the gradients $(\nabla f(x_k))^{(i)}$ for $i \in S_k$, c_2 is the cost of computing the updates $h_k^{(i)}$ for $i \in S_k$, and c_3 is the cost of applying these updates. For simplicity, assume that all blocks are of size 1 and that we update τ blocks at each

iteration. Clearly, $c_3 = \tau$. Since often $h_k^{(i)}$ can be computed in closed form[6] and takes $O(1)$ operations, we have $c_2 = O(\tau)$. The value of c_1 is more difficult to predict in general since by Proposition 2, we have

$$\nabla f_\mu(x_k) = A^T z_k,$$

where $z_k = \arg\max_{z \in Q} \{\langle Ax_k, z\rangle - g(z) - \mu d(z)\}$, and hence c_1 depends on the relationship between A, Q, g and d. It is often the case though that z_{k+1} is obtained from z_k by changing at most δ coordinates, with δ being small. In such a case it is efficient to maintain the vectors $\{z_k\}$ (update at each iteration will cost δ) and at iteration k to compute $(\nabla f_\mu(x_k))^{(i)} = (A^T z_k)^{(i)} = \langle a_i, z_k\rangle$ for $i \in S_k$, where a_i is the i-th column of A, whence $c_1 = \delta + 2\sum_{i \in S_k} \|a_i\|_0$. Since $\mathbf{P}(i \in S_k) = \tau/n$, we have

$$\mathbf{E}[c_1] = \delta + \tfrac{2\tau}{n} \sum_{i=1}^{n} \|a_i\|_0 = \delta + \tfrac{2\tau}{n} \operatorname{nnz}(A).$$

In summary, the expected overall arithmetic cost of a single iteration of SPCDM, under the assumptions made above, is $\mathbf{E}[c] = O(\frac{\tau}{n} \operatorname{nnz}(A) + \delta)$.

6. **Parallel randomized AdaBoost.** We observe that the *logarithm* of the exponential loss function, which is very popular in machine learning[7], is of the form

$$f_\mu(x) = \log\Big(\tfrac{1}{m} \sum_{j=1}^{m} \exp(b_j (Ax)_j)\Big).$$

for $\mu = 1$ and $f(x) = \max_j b_j (Ax)_j$. SPCDM in this case can be interpreted as a parallel randomized boosting method. More details are given in Section 6.3, and in a follow up[8] paper of Fercoq [5]. Our complexity results improve on those in the machine learning literature. Moreover, our framework makes possible the use of regularizers. Note that Nesterov separability in the context of machine learning requires all examples to depend on at most ω features, which is often the case.

7. **Big data friendliness.** Our method is suitable for solving *big data* nonsmooth (7) and smooth (8) convex composite Nesterov separable problems in cases when ω is relatively small compared to n. The reasons for this are: i) the parameters of our method (β and $w = w^*$) can be obtained easily, ii) the cost of a single iteration decreases for smaller ω, iii) the method is equipped with provable parallelization speedup bounds which get better as ω decreases, iv) many real-life big-data problems are sparse and can be modeled in our framework as problems with small ω, v) we demonstrate through numerical experiments involving preliminary medium-scale experiments involving millions of variables that our methods are scalable and that our theoretical parallelization speedup predictions hold.

[6] This is the case in many cases, including i) $\Psi_i(t) = \lambda_i |t|$, ii) $\Psi_i(t) = \lambda_i t^2$, and iii) $\Psi_i(t) = 0$ for $t \in [a_i, b_i]$ and $+\infty$ outside this interval (and the multivariate/block generalizations of these functions). For complicated functions $\Psi_i(t)$ one may need to do one-dimensional optimization, which will cost $O(1)$ for each i, provided that we are happy with an inexact solution. An analysis of PCDM in the $\tau = 1$ case in such an inexact setting can be found in Tappenden et al [36], and can be extended to the parallel setting.

[7] Schapire and Freund have written a book [29] entirely dedicated to boosting and *boosting methods*, which are serial/sequential greedy coordinate descent methods, independently discovered in the machine learning community. The original boosting method, AdaBoost, minimizes the exponential loss, and it the most famous boosting algorithm.

[8] The results presented in this paper were obtained the Fall of 2012 and Spring of 2013, the follow up work of Fercoq [5] was prepared in the Summer of 2013.

8. **Subspace Lipschitz constants.** We derive simple formulas for Lipschitz constants of the gradient of f_μ associated with subspaces spanned by an arbitrary subset S of blocks (Section 3). As a special case, we show that the gradient of a Nesterov separable function is Lipschitz with respect to the norm separable $\|\cdot\|_{w^*}$ with constant equal to $\frac{\omega}{\sigma\mu}$, where ω is degree of Nesterov separability. Besides being useful in our analysis, these results are also of independent interest in the design of gradient-based algorithms in big dimensions.

3 Fast Computation of Subspace Lipschitz Constants

Let us start by introducing the key concept of this section.

Definition 4. Let $\phi:\mathbb{R}^N\to\mathbb{R}$ be a smooth function and let $\emptyset\neq S\subseteq\{1,2,\dots,n\}$. Then we say that $L_S(\nabla\phi)$ is a *Lipschitz constant of $\nabla\phi$ associated with S*, with respect to norm $\|\cdot\|$, if

$$\phi(x+h_{[S]})\leq\phi(x)+\langle\nabla\phi(x),h_{[S]}\rangle+\frac{L_S(\nabla\phi)}{2}\|h_{[S]}\|^2,\qquad x,h\in\mathbb{R}^N. \tag{30}$$

We will alternatively say that $L_S(\nabla\phi)$ is a subspace Lipschitz constant of $\nabla\phi$ *corresponding to the subspace spanned by blocks i for $i\in S$*, that is, $\{\sum_{i\in S}U_ix^{(i)}\ :\ x^{(i)}\in\mathbb{R}^{N_i}\}$, or simply a *subspace Lipschitz constant*.

Observe the above inequality can can be equivalently written as

$$\phi(x+h)\leq\phi(x)+\langle\nabla\phi(x),h\rangle+\frac{L_{\Omega(h)}(\nabla\phi)}{2}\|h\|^2,\qquad x,h\in\mathbb{R}^N.$$

In this section we will be concerned with obtaining easily computable formulas for subspace Lipschitz constants for $\phi=f_\mu$ with respect to the separable norm $\|\cdot\|_w$. Inequalities of this type were first introduced in [26, Section 4] (therein called Deterministic Separable Overapproximation, or DSO). The basic idea is that in a parallel coordinate descent method in which τ blocks are updated at each iteration, subspace Lipschitz constants for sets S of cardinality τ are more relevant (and possibly much smaller = better) than the standard Lipschitz constant of the gradient, which corresponds to the special case $S=\{1,2,\dots,n\}$ in the above definition. This generalizes the concept of block/coordinate Lipschitz constants introduced by Nesterov [19] (in which case $|S|=1$) to spaces spanned by multiple blocks.

We first derive a *generic* bound on subspace Lipschitz constants (Section 3.2), one that holds for any choice of w and v. Subsequently we show (Section 3.3) that for a particular data-dependent choice of the parameters $w_1,\dots,w_n>0$ defining the norm in $\mathbb{R}^N$, the generic bound can be written in a very simple form from which it is clear that i) $L_S\leq L_{S'}$ whenever $S\subset S'$ and ii) that L_S decreases as the degree of Nesterov separability ω decreases. Moreover, it is important that the data-dependent weights w^* and the factor are easily computable, as these parameters are needed to run the algorithm.

3.1 Primal update spaces

As a first step we need to construct a collection of normed spaces associated with the subsets of $\{1,2,\dots,n\}$. These will be needed in the technical proofs and also in the formulation of our results.

- **Spaces.** For $\emptyset \neq S \subseteq \{1,2,\dots,n\}$ we define $\mathbb{R}^S \stackrel{\text{def}}{=} \bigotimes_{i\in S} \mathbb{R}^{N_i}$ and for $h \in \mathbb{R}^N$ we write $h^{(S)}$ for the vector in $\mathbb{R}^S$ obtained from h by deleting all coordinates belonging to blocks $i \notin S$ (and otherwise keeping the order of the coordinates).[9]

- **Matrices.** Likewise, let $A^{(S)} : \mathbb{R}^S \to \mathbb{R}^m$ be the matrix obtained from $A \in \mathbb{R}^{m\times N}$ by deleting all columns corresponding to blocks $i \notin S$, and note that

$$A^{(S)}h^{(S)} = Ah_{[S]}. \tag{31}$$

- **Norms.** We fix positive scalars $w_1, w_2, \dots, w_n$ and on $\mathbb{R}^S$ define a pair of conjugate norms as follows

$$\|h^{(S)}\|_w \stackrel{\text{def}}{=} \left(\sum_{i\in S} w_i \langle B_i h^{(i)}, h^{(i)}\rangle\right)^{1/2}, \qquad \|h^{(S)}\|_w^* \stackrel{\text{def}}{=} \left(\sum_{i\in S} w_i^{-1} \langle B_i^{-1} h^{(i)}, h^{(i)}\rangle\right)^{1/2}. \tag{32}$$

  The standard Euclidean norm of a vector $h^{(S)} \in \mathbb{R}^S$ is given by

$$\|h^{(S)}\|_E^2 = \sum_{i\in S} \|h^{(i)}\|_E^2 = \sum_{i\in S} \langle h^{(i)}, h^{(i)}\rangle. \tag{33}$$

Remark: Note that, in particular, for $S = \{i\}$ we get $h^{(S)} = h^{(i)} \in \mathbb{R}^{N_i}$ and $\mathbb{R}^S \equiv \mathbb{R}^{N_i}$ (primal block space); and for $S = [n]$ we get $h^{(S)} = h \in \mathbb{R}^N$ and $\mathbb{R}^S \equiv \mathbb{R}^N$ (primal basic space). Moreover, for all $\emptyset \neq S \subseteq [n]$ and $h \in \mathbb{R}^N$,

$$\|h^{(S)}\|_w = \|h_{[S]}\|_w, \tag{34}$$

where the first norm is in $\mathbb{R}^S$ and the second in $\mathbb{R}^N$.

3.2 General bound

Our first result in this section, Theorem 5, is a refinement of inequality (27) for a sparse update vector h. The only change consists in the term $\|A\|_{w,v}^2$ being replaced by $\|A^{(S)}\|_{w,v}^2$, where $S = \Omega(h)$ and $A^{(S)}$ is the matrix, defined in Section 3.1, mapping vectors in the primal update space $\mathbb{E}_1 \equiv \mathbb{R}^S$ to vectors in the dual basic space $\mathbb{E}_2 \equiv \mathbb{R}^m$. The primal and dual norms are given by $\|\cdot\|_1 \equiv \|\cdot\|_w$ and $\|\cdot\|_2 \equiv \|\cdot\|_v$, respectively. This is indeed a refinement, since for any $\emptyset \neq S \subseteq [n]$,

$$\begin{aligned}
\|A\|_{w,v} &\stackrel{(22)}{=} \max_{\substack{\|h\|_w=1 \\ h\in\mathbb{R}^N}} \|Ah\|_v^* \\
&\geq \max_{\substack{\|h\|_w=1 \\ h^{(i)}=0,\ i\in S \\ h\in\mathbb{R}^N}} \|Ah\|_v^* \stackrel{(15)}{=} \max_{\substack{\|h_{[S]}\|_w=1 \\ h\in\mathbb{R}^N}} \|Ah_{[S]}\|_v^* \stackrel{(31)+(34)}{=} \max_{\substack{\|h^{(S)}\|_w=1 \\ h\in\mathbb{R}^N}} \|A^{(S)}h^{(S)}\|_v^* \stackrel{(22)}{=} \|A^{(S)}\|_{w,v}.
\end{aligned}$$

The improvement can be dramatic, and gets better for smaller sets S; this will be apparent later. Note that in the same manner one can show that $\|A^{(S_1)}\|_{w,v} \leq \|A^{(S_2)}\|_{w,v}$ if $\emptyset \neq S_1 \subset S_2$.

[9] Note that $h^{(S)}$ is different from $h_{[S]} = \sum_{i\in S} U_i h^{(i)}$, which is a vector in $\mathbb{R}^N$, although both $h^{(S)}$ and $h_{[S]}$ are composed of blocks $h^{(i)}$ for $i \in S$.

Theorem 5 (Subspace Lipschitz Constants)**.** *For any $x \in \mathbb{R}^N$ and nonzero $h \in \mathbb{R}^N$,*

$$f_\mu(x+h) \le f_\mu(x) + \langle \nabla f_\mu(x), h\rangle + \frac{\|A^{(\Omega(h))}\|_{w,v}^2}{2\mu\sigma}\|h\|_w^2. \tag{35}$$

Proof. Fix $x \in \mathbb{R}^N$, $\emptyset \neq S \subseteq [n]$ and define $\bar{f} : \mathbb{R}^S \to \mathbb{R}$ by

$$\begin{aligned} \bar{f}(h^{(S)}) \stackrel{\text{def}}{=} f_\mu(x + h_{[S]}) &= \max_{u\in Q}\left\{\langle A(x+h_{[S]}), u\rangle - g(u) - \mu d(u)\right\} \\ &\stackrel{(31)}{=} \max_{u\in Q}\left\{\langle A^{(S)}h^{(S)}, u\rangle - \bar{g}(u) - \mu d(u)\right\}, \end{aligned} \tag{36}$$

where $\bar{g}(u) = g(u) - \langle Ax, u\rangle$. Applying Proposition 2 (with $\mathbb{E}_1 = \mathbb{R}^S$, $\mathbb{E}_2 = \mathbb{R}^m$, $\bar{A} = A^{(S)}$, $\bar{Q} = Q$, $\|\cdot\|_1 = \|\cdot\|_w$ and $\|\cdot\|_2 = \|\cdot\|_v$), we conclude that the gradient of $\bar{f}$ is Lipschitz with respect to $\|\cdot\|_w$ on $\mathbb{R}^S$, with Lipschitz constant $\frac{1}{\mu\sigma}\|A^{(S)}\|_{w,v}^2$. Hence, for all $h \in \mathbb{R}^N$,

$$f_\mu(x+h_{[S]}) = \bar{f}(h^{(S)}) \le \bar{f}(0) + \langle \nabla\bar{f}(0), h^{(S)}\rangle + \frac{\|A^{(S)}\|_{w,v}^2}{2\mu\sigma}\|h^{(S)}\|_w^2 \ . \tag{37}$$

Note that $\nabla\bar{f}(0) = (A^{(S)})^T u^*$ and $\nabla f_\mu(x) = A^T u^*$, where u^* is the maximizer in (36), whence

$$\langle \nabla\bar{f}(0), h^{(S)}\rangle = \langle (A^{(S)})^T u^*, h^{(S)}\rangle = \langle u^*, A^{(S)}h^{(S)}\rangle \stackrel{(31)}{=} \langle u^*, Ah_{[S]}\rangle = \langle A^T u^*, h_{[S]}\rangle = \langle \nabla f_\mu(x), h_{[S]}\rangle.$$

Substituting this and the identities $\bar{f}(0) = f_\mu(x)$ and (34) into (37) gives

$$f_\mu(x+h_{[S]}) \le f_\mu(x) + \langle \nabla f_\mu(x), h_{[S]}\rangle + \frac{\|A^{(S)}\|_{w,v}^2}{2\mu\sigma}\|h_{[S]}\|_w^2.$$

It now remains to observe that in view of (17) and (15), for all $h \in \mathbb{R}^N$ we have $h_{[\Omega(h)]} = h$. □

3.3 Bounds for data-dependent weights w

From now on we will not consider arbitrary weight vector w but one defined by the data matrix A as follows. Let us define $w^* = (w_1^*, \dots, w_n^*)$ by

$$w_i^* \stackrel{\text{def}}{=} \max\{(\|A_i B_i^{-1/2} t\|_v^*)^2 \ : \ t \in \mathbb{R}^{N_i}, \ \|t\|_E = 1\}, \quad i = 1, 2, \dots, n. \tag{38}$$

Notice that as long as the matrices $A_1, \dots, A_n$ are nonzero, we have $w_i^* > 0$ for all i, and hence the norm $\|\cdot\|_1 = \|\cdot\|_{w^*}$ is well defined. When all blocks are of size 1 (i.e., $N_i = 1$ for all i) and $B_i = 1$ for all i, this reduces to (29). Let us return to the general block setting. Letting $S = \{i\}$ and $\|\cdot\|_1 \equiv \|\cdot\|_{w^*}$, we see that w_i^* is defined so that the $\|A^{(S)}\|_{w^*,v} = 1$. Indeed,

$$\begin{aligned} \|A^{(S)}\|_{w^*,v}^2 &\stackrel{(22)}{=} \max_{\|h^{(S)}\|_{w^*}=1} (\|A^{(S)}h^{(S)}\|_v^*)^2 \stackrel{(31)+(15)}{=} \max_{\|h^{(i)}\|_{w^*}=1} (\|AU_i h^{(i)}\|_v^*)^2 \\ &\stackrel{(16)+(32)}{=} \tfrac{1}{w_i^*} \max_{\|y^{(i)}\|_E=1} (\|AU_i B_i^{-1/2} y^{(i)}\|_v^*)^2 \stackrel{(38)}{=} 1. \end{aligned} \tag{39}$$

In the rest of this section we establish an easily computable upper bound on $\|A^{(\Omega(h))}\|_{w^*,v}^2$ which will be useful in proving a complexity result for SPCDM used with a τ-uniform or τ-nice sampling. The result is, however, of independent interest, as we argue at the end of this section.

The following is a technical lemma needed to establish the main result of this section.

Lemma 6. *For any $\emptyset \neq S \subseteq [n]$ and w^* chosen as in (38), the following hold:*

$$p = 1 \qquad \Rightarrow \qquad \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\leq j\leq m} v_j^{-2} \sum_{i\in S} (A_{ji}h^{(i)})^2 \leq 1,$$

$$1 < p \leq 2 \qquad \Rightarrow \qquad \max_{\|h^{(S)}\|_{w^*}=1} \sum_{j=1}^{m} \left(v_j^{-q} \sum_{i\in S} (A_{ji}h^{(i)})^2 \right)^{q/2} \leq 1.$$

Proof. For any $h^{(i)}$ define the transformed variable $y^{(i)} = (w_i^*)^{1/2} B_i^{1/2} h^{(i)}$ and note that

$$\|h^{(S)}\|_{w^*}^2 \overset{(32)+(16)}{=} \sum_{i\in S} w_i^* \langle B_i h^{(i)}, h^{(i)} \rangle = \sum_{i\in S} \langle y^{(i)}, y^{(i)} \rangle \overset{(33)}{=} \|y^{(S)}\|_E^2.$$

We will now prove the result separately for $p = 1$, $p = 2$ and $1 < p < 2$. For $p = 1$ we have

$$\begin{aligned}
LHS &\overset{\text{def}}{=} \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\leq j\leq m} v_j^{-2} \sum_{i\in S} (A_{ji}h^{(i)})^2 \\
&= \max_{\|y^{(S)}\|_E=1} \max_{1\leq j\leq m} \left(v_j^{-2} \sum_{i\in S} (w_i^*)^{-1} (A_{ji}B_i^{-1/2} y^{(i)})^2 \right) \\
&\leq \max_{\|y^{(S)}\|_E=1} \left(\sum_{i\in S} (w_i^*)^{-1} \max_{1\leq j\leq m} \left(v_j^{-2} (A_{ji}B_i^{-1/2} y^{(i)})^2 \right) \right) \\
&= \max_{\|y^{(S)}\|_E=1} \left(\sum_{i\in S} \|y^{(i)}\|_E^2 \underbrace{(w_i^*)^{-1} \max_{1\leq j\leq m} \left(v_j^{-2} \left(A_{ji}B_i^{-1/2} \tfrac{y^{(i)}}{\|y^{(i)}\|_E} \right)^2 \right)}_{\leq \|A^{(\{i\})}\|_{w^*,v}^2 = 1} \right) \\
&\overset{(39)}{\leq} \max_{\|y^{(S)}\|_E=1} \sum_{i\in S} \|y^{(i)}\|_E^2 \overset{(33)}{=} 1.
\end{aligned}$$

For $p > 1$ we may write:

$$LHS \overset{\text{def}}{=} \max_{\|h^{(S)}\|_{w^*}=1} \sum_{j=1}^{m} v_j^{-q} \left(\sum_{i\in S} (A_{ji}h^{(i)})^2 \right)^{q/2} = \max_{\|y^{(S)}\|_E=1} \sum_{j=1}^{m} v_j^{-q} \left(\sum_{i\in S} (w_i^*)^{-1} (A_{ji}B_i^{-1/2} y^{(i)})^2 \right)^{q/2}. \tag{40}$$

In particular, for $p = 2$ (i.e., $q = 2$) we now have

$$\begin{aligned}
LHS &\overset{(40)}{=} \max_{\|y^{(S)}\|_E=1} \sum_{i\in S} (w_i^*)^{-1} \sum_{j=1}^{m} v_j^{-2} (A_{ji}B_i^{-1/2} y^{(i)})^2 \\
&= \max_{\|y^{(S)}\|_E=1} \sum_{i\in S} \|y^{(i)}\|_E^2 \underbrace{(w_i^*)^{-1} \sum_{j=1}^{m} v_j^{-2} \left(A_{ji}B_i^{-1/2} \tfrac{y^{(i)}}{\|y^{(i)}\|_E} \right)^2}_{\leq \|A^{(\{i\})}\|_{w^*,v}^2 = 1} \\
&\overset{(39)}{\leq} \max_{\|y^{(S)}\|_E=1} \sum_{i\in S} \|y^{(i)}\|_E^2 \overset{(33)}{=} 1.
\end{aligned}$$

For $1 < p < 2$ we will continue[10] from (40), first by bounding $R \overset{\text{def}}{=} \sum_{i\in S}(w_i^*)^{-1}(A_{ji}B_i^{-1/2}y^{(i)})^2$ using the Hölder inequality in the form

$$\sum_{i\in S} a_i b_i \le \left(\sum_{i\in S}|a_i|^s\right)^{1/s}\left(\sum_{i\in S}|b_i|^{s'}\right)^{1/s'},$$

with $a_i = (w_i^*)^{-1}\left(A_{ji}B_i^{-1}\frac{y^{(i)}}{\|y^{(i)}\|_E}\right)^2\|y^{(i)}\|^{2-2/s'}$, $b_i = \|y^{(i)}\|_E^{2/s'}$, $s = q/2$ and $s' = q/(q-2)$.

$$\begin{aligned} R^{q/2} &\le \left(\sum_{i\in S}(w_i^*)^{-q/2}\left|A_{ji}B_i^{-1}\tfrac{y^{(i)}}{\|y^{(i)}\|_E}\right|^q\|y^{(i)}\|_E^2\right)\times\underbrace{\left(\sum_{i\in S}\|y^{(i)}\|_E^2\right)^{(q-2)q/4}}_{\le 1} \\ &\le \sum_{i\in S}(w_i^*)^{-q/2}\left|A_{ji}B_i^{-1}\tfrac{y^{(i)}}{\|y^{(i)}\|_E}\right|^q\|y^{(i)}\|_E^2. \end{aligned} \tag{41}$$

We now substitute (41) into (40) and continue as in the $p = 2$ case:

$$\begin{aligned} LHS &\overset{(40)+(41)}{\le} \max_{\|y^{(S)}\|_E=1}\left(\sum_{j=1}^m v_j^{-q}\sum_{i\in S}(w_i^*)^{-q/2}\left|A_{ji}B_i^{-1}\tfrac{y^{(i)}}{\|y^{(i)}\|_E}\right|^q\|y^{(i)}\|_E^2\right)^{2/q} \\ &= \max_{\|y^{(S)}\|_E=1}\left(\sum_{i\in S}\|y^{(i)}\|_E^2\underbrace{(w_i^*)^{-q/2}\sum_{j=1}^m v_j^{-q}\left|A_{ji}B_i^{-1}\tfrac{y^{(i)}}{\|y^{(i)}\|_E}\right|^q}_{\le\left(\|A^{(\{i\})}\|^2_{w^*,v}\right)^{1/q}\le 1}\right)^{2/q} \\ &\overset{(39)}{\le} \max_{\|y^{(S)}\|_E=1}\left(\sum_{i\in S}\|y^{(i)}\|_E^2\right)^{2/q} = \left(\max_{\|y^{(S)}\|_E=1}\sum_{i\in S}\|y^{(i)}\|_E^2\right)^{2/q} \overset{(33)}{=} 1. \end{aligned}$$

□

Using the above lemma we can now give a simple and easily interpretable bound on $\|A^{(S)}\|^2_{w^*,v}$.

Lemma 7. *For any $\emptyset \ne S \subseteq [n]$ and w^* chosen as in (38),*

$$\|A^{(S)}\|^2_{w^*,v} \le \max_{1\le j\le m}|\Omega(A^T e_j)\cap S|.$$

Proof. It will be useful to note that

$$e_j^T A^{(S)}h^{(S)} \overset{(31)}{=} e_j^T A h_{[S]} \overset{(15)+(18)}{=} \sum_{i\in S}A_{ji}h^{(i)}. \tag{42}$$

[10]The proof works for $p = 2$ as well, but the one we have given for $p = 2$ is simpler, so we included it.

We will (twice) make use the following form of the Cauchy-Schwarz inequality: for scalars a_i, $i \in Z$, we have $(\sum_{i\in Z} a_i)^2 \le |Z| \sum_{i\in Z} a_i^2$. For $p = 1$, we have

$$\begin{aligned}
\|A^{(S)}\|^2_{w^*,v} &\overset{(22)}{=} \max_{\|h^{(S)}\|_{w^*}\le 1} (\|A^{(S)}h^{(S)}\|_v^*)^2 \overset{(12)}{=} \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\le j\le m} v_j^{-2}\left(e_j^T A^{(S)}h^{(S)}\right)^2 \\
&\overset{(42)+(19)}{=} \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\le j\le m} v_j^{-2}\left(\sum_{i\in\Omega(A^Te_j)\cap S} A_{ji}h^{(i)}\right)^2 \\
&\overset{\text{(Cauchy-Schwarz)}}{\le} \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\le j\le m} \left(v_j^{-2}|\Omega(A^Te_j)\cap S| \sum_{i\in\Omega(A^Te_j)\cap S} (A_{ji}h^{(i)})^2\right) \\
&\le \max_{1\le j\le m} |\Omega(A^Te_j)\cap S| \times \max_{\|h^{(S)}\|_{w^*}=1} \max_{1\le j\le m} \left(v_j^{-2}\sum_{i\in S}(A_{ji}h^{(i)})^2\right) \\
&\overset{\text{(Lemma 6)}}{\le} \max_{1\le j\le m} |\Omega(A^Te_j)\cap S|.
\end{aligned}$$

For $1 < p \le 2$, we may write

$$\begin{aligned}
\|A^{(S)}\|^2_{w^*,v} &\overset{(22)}{=} \max_{\|h^{(S)}\|_{w^*}\le 1} (\|A^{(S)}h^{(S)}\|_v^*)^2 \overset{(12)}{=} \max_{\|h^{(S)}\|_{w^*}=1} \left(\sum_{j=1}^m v_j^{-q}\left|e_j^T A^{(S)}h^{(S)}\right|^q\right)^{1/q} \\
&\overset{(42)+(19)}{=} \max_{\|h^{(S)}\|_{w^*}=1} \left(\sum_{j=1}^m v_j^{-q}\left(\left|\sum_{i\in\Omega(A^Te_j)\cap S} A_{ji}h^{(i)}\right|^2\right)^{q/2}\right)^{2/q} \\
&\overset{\text{(Cauchy-Schwarz)}}{\le} \max_{\|h^{(S)}\|_{w^*}=1} \left(\sum_{j=1}^m v_j^{-q}\left(\left|\Omega(A^Te_j)\cap S\right| \sum_{i\in\Omega(A^Te_j)\cap S} (A_{ji}h^{(i)})^2\right)^{q/2}\right)^{2/q} \\
&\le \max_{1\le j\le m} |\Omega(A^Te_j)\cap S| \times \max_{\|h^{(S)}\|_{w^*}=1} \left(\sum_{j=1}^m v_j^{-q}\left(\sum_{i\in S}(A_{ji}h^{(i)})^2\right)^{q/2}\right)^{2/q} \\
&\overset{\text{(Lemma 6)}}{\le} \max_{1\le j\le m} |\Omega(A^Te_j)\cap S|.
\end{aligned}$$

□

We are now ready to state and prove the main result of this section. It says that the (interesting but somewhat non-informative) quantity $\|A^{(\Omega(h))}\|^2_{w,v}$ appearing in Theorem 5 can for $w = w^*$ be bounded by a very natural and easily computable quantity capturing the interplay between the sparsity pattern of the rows of A and the sparsity pattern of h.

Theorem 8 (Subspace Lipschitz Constants for $w = w^*$)**.** *For $S \subseteq \{1, 2, \dots, n\}$ let*

$$L_S \overset{def}{=} \max_{1\le j\le m} |\Omega(A^Te_j)\cap S|. \tag{43}$$

Then for all $x, h \in \mathbb{R}^N$,

$$f_\mu(x+h) \le f_\mu(x) + \langle \nabla f_\mu(x), h\rangle + \frac{L_{\Omega(h)}}{2\mu\sigma}\|h\|_{w^*}^2. \tag{44}$$

Proof. In view of Theorem 5, we only need to show that $\|A^{(\Omega(h))}\|_{w^*,v}^2 \le L_{\Omega(h)}$. This directly follows from Lemma 7. □

Let us now comment on the meaning of this theorem:

1. Note that $L_{\Omega(h)}$ depends on A and h through their *sparsity pattern* only. Furthermore, μ is a user chosen parameter and σ depends on d and the choice of the norm $\|\cdot\|_v$, which is independent of the data matrix A. Hence, the term $\frac{L_{\Omega(h)}}{\mu\sigma}$ is *independent* of the *values* of A and h. Dependence on A is entirely contained in the weight vector w^*, as defined in (38).

2. For each S we have $L_S \le \min\{\max_{1\le j\le m}|\Omega(A^T e_j)|, |S|\} = \min\{\omega, |S|\} \le \omega$, where ω is the degree of Nesterov separability of f.

   (a) By substituting the bound $L_S \le \omega$ into (44) we conclude that the gradient of f_μ is Lipschitz with respect to the norm $\|\cdot\|_{w^*}$, with Lipschitz constant equal to $\frac{\omega}{\mu\sigma}$.

   (b) By substituting $U_i h^{(i)}$ in place of h in (44) (we can also use Theorem 5), we observe that the gradient of f_μ is *block Lipschitz* with respect to the norm $\langle B_i\cdot,\cdot\rangle^{1/2}$, with Lipschitz constant corresponding to block i equal to $L_i = \frac{w_i^*}{\mu\sigma}$:

$$f_\mu(x+U_i h^{(i)}) \le f_\mu(x) + \langle \nabla f_\mu(x), U_i h^{(i)}\rangle + \frac{L_i}{2}\langle B_i h^{(i)}, h^{(i)}\rangle, \quad x\in\mathbb{R}^N,\ h^{(i)}\in\mathbb{R}^{N_i}.$$

3. In some sense it is more natural to use the norm $\|\cdot\|_L^2$ instead of $\|\cdot\|_{w^*}^2$, where $L = (L_1,\dots,L_n)$ are the block Lipschitz constants $L_i = \frac{w_i^*}{\mu\sigma}$ of ∇f_μ. If we do this, then although the situation is very different, inequality (44) is similar to the one given for partially separable smooth functions in [26, Theorem 7]. Indeed, the weights defining the norm are in both cases equal to the block Lipschitz constants (of f in [26] and of f_μ here). Moreover, the leading term in [26] is structurally comparable to the leading term $L_{\Omega(h)}$. Indeed, it is equal to $\max_S|\Omega(h)\cap S|$, where the maximum is taken over the block domains S of the constituent functions $f_S(x)$ in the representation of f revealing partial separability: $f(x) = \sum_S f_S(x)$.

4 Expected Separable Overapproximation (ESO)

In this section we compute parameters β and w yielding an ESO for the pair $(\phi, \hat{S})$, where $\phi = f_\mu$ and $\hat{S}$ is a proper uniform sampling. If inequality (3) holds, we will for simplicity write $(\phi,\hat{S}) \sim \text{ESO}(\beta, w)$. Note also that for all $\gamma > 0$,

$$(\phi,\hat{S}) \sim \text{ESO}(\beta\gamma, w) \qquad \Longleftrightarrow \qquad (\phi,\hat{S}) \sim \text{ESO}(\beta, \gamma w).$$

In Section 4.1 we establish a link between ESO for $(\phi,\hat{S})$ and Lipschitz continuity of the gradient of a certain collection of functions. This link will enable us to compute the ESO parameters β, w for the smooth approximation of a Nesterov separable function f_μ, needed both for running

Algorithm 1 and for the complexity analysis. In Section 4.2 we define certain technical objects that will be needed for further analysis. In Section 4.3 we prove a first ESO result, computing β for any $w > 0$ and *any* proper uniform sampling. The formula for β involves the norm of a certain large matrix, and hence is not directly useful as β is needed for running the algorithm. Also, this formula does not explicitly exhibit dependence on ω; that is, it is not immediately apparent that β will be smaller for smaller ω, as one would expect. Subsequently, in Section **??**, we specialize this result to τ-uniform samplings and then further to the more-specialized τ-nice samplings in Section **??**. As in the previous section, in these special cases we show that the choice $w = w^*$ leads to very simple closed-form expressions for β, allowing us to get direct insight into parallelization speedup.

4.1 ESO and Lipschitz continuity

We will now study the collection of functions $\hat{\phi}_x : \mathbb{R}^N \to \mathbb{R}$ for $x \in \mathbb{R}^N$ defined by

$$\hat{\phi}_x(h) \stackrel{\text{def}}{=} \mathbf{E}\left[\phi(x + h_{[\hat{S}]})\right]. \tag{45}$$

Let us first establish some basic connections between ϕ and $\hat{\phi}_x$.

Lemma 9. *Let $\hat{S}$ be any sampling and $\phi : \mathbb{R}^N \to \mathbb{R}$ any function. Then for all $x \in \mathbb{R}^N$*

(i) if ϕ is convex, so is $\hat{\phi}_x$,

(ii) $\hat{\phi}_x(0) = \phi(x)$,

(iii) If $\hat{S}$ is proper and uniform, and $\phi : \mathbb{R}^N \to \mathbb{R}$ is continuously differentiable, then

$$\nabla\hat{\phi}_x(0) = \frac{\mathbf{E}[|\hat{S}|]}{n}\nabla\phi(x).$$

Proof. Fix $x \in \mathbb{R}^N$. Notice that $\hat{\phi}_x(h) = \mathbf{E}[\phi(x + h_{[\hat{S}]})] = \sum_{S\subseteq[n]} \mathbf{P}(\hat{S} = S)\phi(x + U_S h)$, where $U_S \stackrel{\text{def}}{=} \sum_{i\in S} U_i U_i^T$. As $\hat{\phi}_x$ is a convex combination of convex functions, it is convex, establishing (i). Property (ii) is trivial. Finally,

$$\nabla\hat{\phi}_x(0) = \mathbf{E}\left[\nabla\ \phi(x + h_{[\hat{S}]})\Big|_{h=0}\right] = \mathbf{E}\left[U_{\hat{S}}\nabla\phi(x)\right] = \mathbf{E}\left[U_{\hat{S}}\right]\nabla\phi(x) = \frac{\mathbf{E}[|\hat{S}|]}{n}\nabla\phi(x).$$

The last equality follows from the observation that $U_{\hat{S}}$ is an $N \times N$ binary diagonal matrix with ones in positions (i, i) for $i \in \hat{S}$ only, coupled with (2). □

We now establish a connection between ESO and a uniform bound in x on the Lipschitz constants of the gradient "at the origin" of the functions $\{\hat{\phi}_x,\ x \in \mathbb{R}^N\}$. The result will be used for the computation of the parameters of ESO for Nesterov separable functions.

Theorem 10. *Let $\hat{S}$ be proper and uniform, and $\phi : \mathbb{R}^N \to \mathbb{R}$ be continuously differentiable. Then the following statements are equivalent:*

(i) $(\phi, \hat{S}) \sim \text{ESO}(\beta, w)$,

(ii) $\hat{\phi}_x(h) \le \hat{\phi}_x(0) + \langle\nabla\hat{\phi}_x(0), h\rangle + \frac{1}{2}\frac{\mathbf{E}[|\hat{S}|]\beta}{n}\|h\|_w^2, \qquad x, h \in \mathbb{R}^N.$

Proof. We only need to substitute (45) and Lemma 9(ii-iii) into inequality (ii) and compare the result with (3). □

4.2 Dual product space

Here we construct a linear space associated with a fixed block sampling $\hat{S}$, and several derived objects which will depend on the distribution of $\hat{S}$. These objects will be needed in the proof of Theorem 11 and in further text.

- **Space.** Let $\mathcal{P} \stackrel{\text{def}}{=} \{S \subseteq [n] \ : \ p_S > 0\}$, where $p_S \stackrel{\text{def}}{=} \mathbf{P}(\hat{S} = S)$. The *dual product space* associated with $\hat{S}$ is defined by
$$\mathbb{R}^{|\mathcal{P}|m} \stackrel{\text{def}}{=} \bigotimes_{S \in \mathcal{P}} \mathbb{R}^m.$$

- **Norms.** Letting $u = \{u^S \in \mathbb{R}^m \ : \ S \in \mathcal{P}\} \in \mathbb{R}^{|\mathcal{P}|m}$, we now define a pair of conjugate norms in $\mathbb{R}^{|\mathcal{P}|m}$ associated with v and $\hat{S}$:
$$\|u\|_{\hat{v}} \stackrel{\text{def}}{=} \Big(\sum_{S \in \mathcal{P}} p_S \|u^S\|_v^2 \Big)^{1/2}, \qquad \|u\|_{\hat{v}}^* \stackrel{\text{def}}{=} \max_{\|u'\|_{\hat{v}} \leq 1} \langle u', u \rangle = \Big(\sum_{S \in \mathcal{P}} p_S^{-1} (\|u^S\|_v^*)^2 \Big)^{1/2}. \tag{46}$$
The notation $\hat{v}$ indicates dependence on both v and $\hat{S}$.

- **Matrices.** For each $S \in \mathcal{P}$ let
$$\hat{A}^S \stackrel{\text{def}}{=} p_S A \sum_{i \in S} U_i U_i^T \in \mathbb{R}^{m \times N}. \tag{47}$$
We now define matrix $\hat{A} \in \mathbb{R}^{|\mathcal{P}|m \times N}$, obtained by stacking the matrices $\hat{A}^S$, $S \in \mathcal{P}$, on top of each other (in the same order the vectors u^S, $S \in \mathcal{P}$ are stacked to form $u \in \mathbb{R}^{|\mathcal{P}|m}$). The "hat" notation indicates that $\hat{A}$ depends on both A and $\hat{S}$. Note that $\hat{A}$ maps vectors from the primal basic space $\mathbb{E}_1 \equiv \mathbb{R}^N$ to vectors in the dual product space $\mathbb{E}_2 \equiv \mathbb{R}^{|\mathcal{P}|m}$. We use $\|\cdot\|_1 \equiv \|\cdot\|_w$ as the norm in $\mathbb{E}_1$ and $\|\cdot\|_2 \equiv \|\cdot\|_{\hat{v}}$ as the norm in $\mathbb{E}_2$. It will be useful to note that for $h \in \mathbb{R}^N$, and $S \in \mathcal{P}$,
$$(\hat{A}h)^S = \hat{A}^S h. \tag{48}$$

4.3 Generic ESO for proper uniform samplings

Our first ESO result covers all (proper) uniform samplings and is valid for any $w > 0$. We give three formulas with three different values of β. While we could have instead given a single formula with β being the minimum of the three values, this will be useful.

Theorem 11 (Generic ESO)**.** *If $\hat{S}$ is proper and uniform, then*
$$i) \quad (f_\mu, \hat{S}) \sim \text{ESO}\left(\frac{n\|\hat{A}\|_{w,\hat{v}}^2}{\mu\sigma \mathbf{E}[|\hat{S}|]}, w \right), \qquad ii) \quad (f_\mu, \hat{S}) \sim \text{ESO}\left(\frac{n\mathbf{E}\left[\|A^{(\hat{S})}\|_{w,v}^2\right]}{\mu\sigma \mathbf{E}[|\hat{S}|]}, w \right), \tag{49}$$
$$iii) \quad (f_\mu, \hat{S}) \sim \text{ESO}\left(\frac{\max_{S \in \mathcal{P}} \|A^{(S)}\|_{w,v}^2}{\mu\sigma}, w \right). \tag{50}$$

Proof. We will first establish (i). Consider the function

$$\begin{aligned}\bar{f}(h) &\overset{\text{def}}{=} \mathbf{E}[f_\mu(x+h_{[\hat{S}]})] \\ &\overset{(23)}{=} \sum_{S\in\mathcal{P}} p_S \max_{u^S\in Q} \left\{\langle A(x+h_{[S]}), u^S\rangle - g(u^S) - \mu d(u^S)\right\} \\ &= \max_{\{u^S\in Q\,:\,S\in\mathcal{P}\}} \sum_{S\in\mathcal{P}} p_S \left\{\langle Ah_{[S]}, u^S\rangle + \langle Ax, u^S\rangle - g(u^S) - \mu d(u^S)\right\}. \end{aligned} \tag{51}$$

Let $u \in \bar{Q} \overset{\text{def}}{=} Q^{|\mathcal{P}|} \subseteq \mathbb{R}^{|\mathcal{P}|m}$ and note that

$$\sum_{S\in\mathcal{P}} p_S \langle Ah_{[S]}, u^S\rangle \overset{(47)+(15)}{=} \sum_{S\in\mathcal{P}} \langle \hat{A}^S h, u^S\rangle \overset{(48)}{=} \langle \hat{A}h, u\rangle. \tag{52}$$

Furthermore, define $\bar{g} : \bar{Q} \to \mathbb{R}$ by $\bar{g}(u) \overset{\text{def}}{=} \sum_{S\in\mathcal{P}} p_S(g(u^S) - \langle Ax, u^S\rangle)$, and $\bar{d} : \bar{Q} \to \mathbb{R}$ by $\bar{d}(u) \overset{\text{def}}{=} \sum_{S\in\mathcal{P}} p_S d(u^S)$. Plugging all of the above into (51) gives

$$\bar{f}(h) = \max_{u\in\bar{Q}} \left\{\langle \hat{A}h, u\rangle - \bar{g}(u) - \mu\bar{d}(u)\right\}. \tag{53}$$

It is easy to see that $\bar{d}$ is σ-strongly convex on $\bar{Q}$ with respect to the norm $\|\cdot\|_{\hat{v}}$ defined in (46). Indeed, for any $u_1, u_2 \in \bar{Q}$ and $t \in (0,1)$,

$$\begin{aligned}\bar{d}(tu_1 + (1-t)u_2) &= \sum_{S\in\mathcal{P}} p_S d(tu_1^S + (1-t)u_2^S) \\ &\leq \sum_{S\in\mathcal{P}} p_S\big(td(u_1^S) + (1-t)d(u_2^S) - \frac{\sigma}{2}t(1-t)\|u_1^S - u_2^S\|_v^2\big) \\ &\overset{(46)}{=} t\bar{d}(u_1) + (1-t)\bar{d}(u_2) - \frac{\sigma}{2}t(1-t)\|u_1 - u_2\|_{\hat{v}}^2.\end{aligned}$$

Due to $\bar{f}$ taking on the form (53), Proposition 2 (used with $\mathbb{E}_1 = \mathbb{R}^N$, $\mathbb{E}_2 = \mathbb{R}^{|\mathcal{P}|m}$, $\bar{A} = \hat{A}$, $\|\cdot\|_1 = \|\cdot\|_w$, $\|\cdot\|_2 = \|\cdot\|_{\hat{v}}$ and $\bar{\sigma} = \sigma$) says that the gradient of $\bar{f}$ is Lipschitz with constant $\frac{1}{\mu\sigma}\|\hat{A}\|_{w,\hat{v}}^2$. We now only need to applying Theorem 10, establishing (i).

Let us now show (ii)+(iii). Fix $h \in \mathbb{R}^N$, apply Theorem 5 with $h \leftarrow h_{[\hat{S}]}$ and take expectations. Using identities (6), we get

$$\mathbf{E}[f_\mu(x+h_{[\hat{S}]})] \leq f(x) + \tfrac{\mathbf{E}[|\hat{S}|]}{n}\left(\langle\nabla f_\mu(x), h\rangle + \tfrac{n\gamma(h)}{2\mu\sigma\mathbf{E}[|\hat{S}|]}\right), \quad \gamma(h) = \mathbf{E}\left[\|A^{(\hat{S})}\|_{w,v}^2 \|h_{[\hat{S}]}\|_w^2\right].$$

Since $\|h_{[\hat{S}]}\|_w^2 \leq \|h\|_w^2$, we have $\gamma(h) \leq \mathbf{E}\left[\|A^{(\hat{S})}\|_{w,v}^2\right]\|h\|_w^2$, which establishes (ii). Since $\|A^{(\hat{S})}\|_{w,v}^2 \leq \max_{S\in\mathcal{P}}\|A^{(S)}\|_{w,v}^2$, using (6) we obtain $\gamma(h) \leq \frac{\mathbf{E}[|\hat{S}|]\max_{S\in\mathcal{P}}\|A^{(S)}\|_{w,v}^2}{n}\|h\|_w^2$, establishing (iii). □

We now give an insightful characterization of $\|\hat{A}\|_{w,\hat{v}}$.

Theorem 12. *If $\hat{S}$ is proper and uniform, then*

$$\|\hat{A}\|_{w,\hat{v}}^2 = \max_{h\in\mathbb{R}^N,\ \|h\|_w\leq 1} \mathbf{E}\left[\left(\|Ah_{[\hat{S}]}\|_v^*\right)^2\right]. \tag{54}$$

Moreover,

$$\left(\frac{\mathbf{E}[|\hat{S}|]}{n}\right)^2 \|A\|_{w,v}^2 \;\le\; \|\hat{A}\|_{w,\hat{v}}^2 \;\le\; \min\left\{\mathbf{E}\left[\|A^{(S)}\|_{w,v}^2\right],\; \frac{\mathbf{E}[|\hat{S}|]}{n}\|A\|_{w,v}^2,\; \max_{S\in\mathcal{P}}\|A^{(S)}\|_{w,v}^2\right\}.$$

Proof. Identity (54) follows from

$$\begin{aligned}
\|\hat{A}\|_{w,\hat{v}} &\overset{(22)}{=} \max\{\langle \hat{A}h, u\rangle \;:\; \|h\|_w \le 1, \quad \|u\|_{\hat{v}} \le 1\} \\
&\overset{(52)+(46)}{=} \max\left\{\sum_{S\in\mathcal{P}} p_S\langle Ah_{[S]}, u^S\rangle \;:\; \|h\|_w \le 1, \quad \sum_{S\in\mathcal{P}} p_S\|u^S\|_v^2 \le 1\right\}. \qquad (55)\\
&= \max_{\|h\|_w\le 1}\max_u\left\{\sum_{S\in\mathcal{P}} p_S\|u^S\|_v\langle Ah_{[S]}, \tfrac{u^S}{\|u^S\|_v}\rangle \;:\; \sum_{S\in\mathcal{P}} p_S\|u^S\|_v^2 \le 1\right\} \\
&= \max_{\|h\|_w\le 1}\max_\beta\left\{\sum_{S\in\mathcal{P}} p_S\beta_S\|Ah_{[S]}\|_v^* \;:\; \sum_{S\in\mathcal{P}} p_S\beta_S^2 \le 1,\; \beta_S\ge 0\right\} \\
&= \max_{\|h\|_w\le 1}\left(\sum_{S\in\mathcal{P}} p_S\left(\|Ah_{[S]}\|_v^*\right)^2\right)^{1/2} \quad = \quad \max_{\|h\|_w\le 1}\left(\mathbf{E}\left[\left(\|Ah_{[\hat{S}]}\|_v^*\right)^2\right]\right)^{1/2}.
\end{aligned}$$

As a consequence, we now have

$$\begin{aligned}
\|\hat{A}\|_{w,\hat{v}}^2 &\overset{(54)}{\le} \mathbf{E}\left[\max_{h\in\mathbb{R}^N,\,\|h\|_w\le 1}\left(\|Ah_{[\hat{S}]}\|_v^*\right)^2\right] \\
&\overset{(31)}{=} \mathbf{E}\left[\max_{h\in\mathbb{R}^N,\,\|h\|_w\le 1}\left(\|A^{(\hat{S})}h^{(\hat{S})}\|_v^*\right)^2\right] \quad \overset{(22)}{=} \quad \mathbf{E}\left[\|A^{(\hat{S})}\|_{w,v}^2\right] \quad \le \quad \max_{S\in\mathcal{P}}\|A^{(S)}\|_{w,v}^2,
\end{aligned}$$

and

$$\|\hat{A}\|_{w,\hat{v}}^2 \overset{(54)}{\le} \max_{h\in\mathbb{R}^N,\,\|h\|_w\le 1}\mathbf{E}\left[\|A\|_{w,v}^2\|h_{[\hat{S}]}\|_w^2\right] \overset{(6)}{=} \frac{\mathbf{E}[|\hat{S}|]}{n}\|A\|_{w,v}^2.$$

Finally, restricting the vectors $\hat{u}^S$, $S\in\mathcal{P}$, to be equal (to z), we obtain the estimate

$$\begin{aligned}
\|\hat{A}\|_{w,\hat{v}} &\overset{(55)}{\ge} \max\{\mathbf{E}[\langle Ah_{[\hat{S}]}, z\rangle] \;:\; \|h\|_w \le 1, \quad \|z\|_v \le 1\} \\
&\overset{(6)}{=} \max\{\tfrac{\mathbf{E}[|\hat{S}|]}{n}\langle Ah, z\rangle \;:\; \|h\|_w \le 1, \quad \|z\|_v \le 1\} \quad \overset{(22)}{=} \quad \frac{\mathbf{E}[|\hat{S}|]}{n}\|A\|_{w,v},
\end{aligned}$$

giving the lower bound. □

Observe that as a consequence of this result, ESO (i) in Theorem 11 is always preferable to ESO (ii). In the following section we will utilize ESO (i) for τ-nice samplings and ESO (iii) for the more general τ-uniform samplings. In particular, we give easily computable upper bounds on β in the special case when $w = w^*$.

4.4 ESO for data-dependent weights w

Let us first establish ESO for τ-uniform samplings and $w = w^*$.

Theorem 13 (ESO for τ-uniform sampling)**.** *If f is Nesterov separable of degree ω, $\hat{S}$ is a τ-uniform sampling and w^* is chosen as in* (38)*, then*

$$(f_\mu, \hat{S}) \sim \text{ESO}\left(\beta, w^*\right),$$

where $\beta = \frac{\beta_1'}{\mu\sigma}$ and $\beta_1' \overset{def}{=} \min\{\omega, \tau\}$.

Proof. This follows from ESO (iii) in Theorem 11 in by using the bound $\|A^{(S)}\|_{w,v}^2 \leq \max_j |\Omega(A^T e_j) \cap S| \leq \min\{\omega, \tau\}$, $S \in \mathcal{P}$, which follows from Lemma 7 and the fact that $|\Omega(A^T e_j)| \leq \omega$ for all j and $|S| = \tau$ for all $S \in \mathcal{P}$. □

Before we establish an ESO result for τ-nice samplings, the main result of this section, we need a technical lemma with a number of useful relations. Identities (57) and (60) and estimate (61) are new, the other two identities are from [26, Section 3]. For $S \subseteq [n] = \{1, 2, \ldots, n\}$ define

$$\chi_{(i \in S)} = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{56}$$

Lemma 14. *Let $\hat{S}$ be any sampling, J_1, J_2 be nonempty subsets of $[n]$ and $\{\theta_{ij} : i \in [n],\ j \in [n]\}$ be any real constants. Then*

$$\mathbf{E}\left[\sum_{i \in J_1 \cap \hat{S}} \sum_{j \in J_2 \cap \hat{S}} \theta_{ij}\right] = \sum_{i \in J_1} \sum_{j \in J_2} \mathbf{P}(\{i, j\} \subseteq \hat{S}) \theta_{ij}. \tag{57}$$

If $\hat{S}$ is τ-nice, then for any $\emptyset \neq J \subseteq [n]$, $\theta \in \mathbb{R}^n$ and $k \in [n]$, the following identities hold

$$\mathbf{E}\left[\sum_{i \in J \cap \hat{S}} \theta_i \mid |J \cap \hat{S}| = k\right] = \frac{k}{|J|} \sum_{i \in J} \theta_i, \tag{58}$$

$$\mathbf{E}\left[|J \cap \hat{S}|^2\right] = \frac{|J|\tau}{n}\Big(1 + \frac{(|J| - 1)(\tau - 1)}{\max(1, n - 1)}\Big), \tag{59}$$

$$\max_{1 \leq i \leq n} \mathbf{E}[|J \cap \hat{S}| \times \chi_{(i \in \hat{S})}] = \frac{\tau}{n}\left(1 + \frac{(|J| - 1)(\tau - 1)}{\max(1, n - 1)}\right). \tag{60}$$

Moreover, if $J_1, \ldots, J_m$ are subsets of $[n]$ of identical cardinality ($|J_j| = \omega$ for all j), then

$$\max_{1 \leq i \leq n} \mathbf{E}[\max_{1 \leq j \leq m} |J_j \cap \hat{S}| \times \chi_{(i \in \hat{S})}] \leq \frac{\tau}{n} \sum_{k=1}^{k_{max}} \min\left\{1\ ,\ \frac{mn}{\tau} \sum_{l = \max\{k, k_{min}\}}^{k_{max}} c_l \pi_l\right\}, \tag{61}$$

where $k_{min} = \max\{1, \tau - (n - \omega)\}$, $k_{max} = \min\{\tau, \omega\}$, $c_l = \max\left\{\frac{l}{\omega}, \frac{\tau - l}{n - \omega}\right\} \leq 1$ if $\omega < n$ and $c_l = \frac{l}{\omega} \leq 1$ otherwise, and

$$\pi_l \overset{def}{=} \mathbf{P}(|J_j \cap \hat{S}| = l) = \frac{\binom{\omega}{k}\binom{n - \omega}{\tau - k}}{\binom{n}{\tau}}, \qquad k_{min} \leq l \leq k_{max}.$$

Proof. The first statement is a straightforward generalization of (26) in [26]. Identities (58) and (59) were established[11] in [26]. Let us prove (60). The statement is trivial for $n = 1$, assume therefore that $n \geq 2$. Notice that

$$\mathbf{E}[|J \cap \hat{S}| \times \chi_{(k\in\hat{S})}] = \mathbf{E}\left[\sum_{i\in J\cap\hat{S}} \sum_{j\in\{k\}\cap\hat{S}} 1\right] \overset{(57)}{=} \sum_{i\in J} \mathbf{P}(\{i,k\} \subseteq \hat{S}). \tag{62}$$

Using (2), and the simple fact that $\mathbf{P}(\{i,k\} \subseteq \hat{S}) = \frac{\tau(\tau-1)}{n(n-1)}$ whenever $i \neq k$, we get

$$\sum_{i\in J} \mathbf{P}(\{i,k\} \subseteq \hat{S}) = \begin{cases} \sum_{i\in J} \frac{\tau(\tau-1)}{n\max(1,n-1)} = \frac{|J|\tau(\tau-1)}{n(n-1)}, & \text{if } k \notin J, \\ \frac{\tau}{n} + \sum_{i\in J/\{k\}} \frac{\tau(\tau-1)}{n(n-1)} = \frac{\tau}{n}\left(1 + \frac{(|J|-1)(\tau-1)}{(n-1)}\right), & \text{if } k \in J. \end{cases} \tag{63}$$

Notice that the expression in the $k \notin J$ case is smaller than expression in the $k \in J$ case. If we now combine (62) and (63) and take maximum in k, (60) is proved. Let us now establish (61). Fix i and let $\eta_j \overset{\text{def}}{=} |J_j \cap \hat{S}|$. We can now estimate

$$\begin{aligned} \mathbf{E}[\max_{1\leq j\leq m} \eta_j \times \chi_{(i\in\hat{S})}] &= \sum_{k=k_{\min}}^{k_{\max}} k\mathbf{P}\left(\max_{1\leq j\leq m} \eta_j \times \chi_{(i\in\hat{S})} = k\right) \\ &= \sum_{k=1}^{k_{\max}} \mathbf{P}\left(\max_{1\leq j\leq m} \eta_j \times \chi_{(i\in\hat{S})} \geq k\right) \\ &= \sum_{k=1}^{k_{\max}} \mathbf{P}\left(\bigcup_{j=1}^{m} \left\{\eta_j \geq k \ \ \& \ \ i \in \hat{S}\right\}\right) \\ &\leq \sum_{k=1}^{k_{\max}} \min\left\{\mathbf{P}(i \in \hat{S}), \sum_{j=1}^{m} \mathbf{P}\left(\eta_j \geq k \ \ \& \ \ i \in \hat{S}\right)\right\} \\ &\overset{(2)}{=} \sum_{k=1}^{k_{\max}} \min\left\{\frac{\tau}{n}, \sum_{j=1}^{m} \sum_{l=\max\{k,k_{\min}\}}^{k_{\max}} \mathbf{P}\left(\eta_j = l \ \ \& \ \ i \in \hat{S}\right)\right\}. \end{aligned} \tag{64}$$

In the last step we have used the fact that $\mathbf{P}(\eta_j = l) = 0$ for $l < k_{\min}$ to restrict the scope of l. Let us now also fix j and estimate $\mathbf{P}(\eta_j = l \ \ \& \ \ i \in \hat{S})$. Consider two cases:

(i) If $i \in J_j$, then among the $\binom{n}{\tau}$ equiprobable possible outcomes of the τ-nice sampling $\hat{S}$, the ones for which $|J_j \cap \hat{S}| = l$ and $i \in \hat{S}$ are those that select block i and $l-1$ other blocks from J_j ($\binom{\omega-1}{l-1}$ possible choices) and $\tau - l$ blocks from outside J_j ($\binom{n-\omega}{\tau-l}$ possible choices). Hence,

$$\mathbf{P}\left(\eta_j = l \ \ \& \ \ i \in \hat{S}\right) = \frac{\binom{\omega-1}{l-1}\binom{n-\omega}{\tau-l}}{\binom{n}{\tau}} = \frac{l}{\omega}\pi_l. \tag{65}$$

[11] In fact, the proof of the former is essentially identical to the proof of (57), and (59) follows from (57) by choosing $J_1 = J_2 = J$ and $\theta_{ij} = 1$.

(ii) If $i \notin J_j$ (notice that this can not happen if $\omega = n$), then among the $\binom{n}{\tau}$ equiprobable possible outcomes of the τ-nice sampling $\hat{S}$, the ones for which $|\hat{S} \cap J_j| = l$ and $i \in \hat{S}$ are those that select block i and $\tau - l - 1$ other blocks from outside J_j ($\binom{n-\omega-1}{\tau-l-1}$ possible choices) and l blocks from J_j ($\binom{\omega}{l}$ possible choices). Hence,

$$\mathbf{P}\left(\eta_j = l \ \ \& \ \ i \in \hat{S}\right) = \frac{\binom{\omega}{l}\binom{n-\omega-1}{\tau-l-1}}{\binom{n}{\tau}} = \frac{\tau - l}{n - \omega}\pi_l. \tag{66}$$

It only remains to plug the maximum of (65) and (66) into (64). □

We are now ready to present the main result of this section.

Theorem 15 (ESO for τ-nice sampling)**.** *Let f be Nesterov separable of degree ω, $\hat{S}$ be τ-nice, and w^* be chosen as in* (38)*. Then*

$$(f_\mu, \hat{S}) \sim \text{ESO}(\beta, w^*),$$

where $\beta = \frac{\beta'}{\mu\sigma}$ and

$$\beta' = \beta'_2 \overset{def}{=} 1 + \frac{(\omega - 1)(\tau - 1)}{\max(1, n - 1)} \tag{67}$$

if the dual norm $\|\cdot\|_v$ is defined with $p = 2$, and

$$\beta' = \beta'_3 \overset{def}{=} \sum_{k=1}^{k_{max}} \min\left\{1, \frac{mn}{\tau} \sum_{l=\max\{k,k_{min}\}}^{k_{max}} c_l \pi_l\right\} \tag{68}$$

if $p = 1$, where c_l, π_l, k_{min} and k_{max} are as in Lemma 14.

Proof. In view of Theorem 11, we only need to bound $\|\hat{A}\|^2_{w^*,\hat{v}}$. First, note that

$$\|\hat{A}\|^2_{w^*,\hat{v}} \overset{(22)}{=} \max_{\|h\|_{w^*}=1} (\|\hat{A}h\|^*_{\hat{v}})^2 \overset{(46)+(48)}{=} \max_{\|h\|_{w^*}=1} \sum_{S \in \mathcal{P}} p_S^{-1} (\|\hat{A}^S h\|^*_v)^2. \tag{69}$$

Further, it will be useful to observe that

$$\hat{A}^S_{ji} \overset{(18)+(47)}{=} p_S e_j^T A \sum_{k \in S} U_k U_k^T U_i \overset{(14)+(18)+(56)}{=} p_S \chi_{(i \in S)} A_{ji} \ . \tag{70}$$

For brevity, let us write $\eta_j \overset{\text{def}}{=} |\Omega(A^T e_j) \cap \hat{S}|$. As $\hat{S}$ is τ-nice, adding dummy dependencies if necessary, we can wlog assume that all rows of A have the same number of nonzero blocks: $|\Omega(A^T e_j)| = \omega$ for all j. Thus, $\pi_k \overset{\text{def}}{=} \mathbf{P}(\eta_j = k)$ does not depend on j. Consider now two cases, depending on whether the norm $\|\cdot\|_v$ in $\mathbb{R}^m$ is defined with $p = 1$ or $p = 2$.

(i) For $p = 2$ we can write

$$\begin{aligned}
\|\hat{A}\|^2_{w^*,\hat{v}} &\overset{(69)+(12)+(42)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S^{-1} \sum_{j=1}^{m} v_j^{-2} \left(\sum_{i=1}^{n} \hat{A}^S_{ji} h^{(i)} \right)^2 \\
&\overset{(70)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S^{-1} \sum_{j=1}^{m} v_j^{-2} \left(\sum_{i=1}^{n} p_S \chi_{(i\in S)} A_{ji} h^{(i)} \right)^2 \\
&\overset{(19)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S \sum_{j=1}^{m} v_j^{-2} \Big(\sum_{i\in\Omega(A^T e_j)\cap S} A_{ji} h^{(i)} \Big)^2 \\
&= \max_{\|h\|_{w^*}=1} \mathbf{E} \left[\sum_{j=1}^{m} v_j^{-2} \Big(\sum_{i\in\Omega(A^T e_j)\cap \hat{S}} A_{ji} h^{(i)} \Big)^2 \right] \\
&= \max_{\|h\|_{w^*}=1} \sum_{k=0}^{n} \mathbf{E} \left[\sum_{j=1}^{m} v_j^{-2} \Big(\sum_{i\in\Omega(A^T e_j)\cap \hat{S}} A_{ji} h^{(i)} \Big)^2 \,\Big|\, \eta_j = k \right] \pi_k \\
&= \max_{\|h\|_{w^*}=1} \sum_{k=0}^{n} \sum_{j=1}^{m} v_j^{-2} \mathbf{E} \left[\Big(\sum_{i\in\Omega(A^T e_j)\cap \hat{S}} A_{ji} h^{(i)} \Big)^2 \,\Big|\, \eta_j = k \right] \pi_k. \quad (71)
\end{aligned}$$

Using the Cauchy-Schwarz inequality, we can write

$$\begin{aligned}
\mathbf{E} \left[\Big(\sum_{i\in\Omega(A^T e_j)\cap \hat{S}} A_{ji} h^{(i)} \Big)^2 \,\Big|\, \eta_j = k \right] &\overset{(\mathrm{CS})}{\leq} \mathbf{E} \left[|\Omega(A^T e_j) \cap \hat{S}| \sum_{i\in\Omega(A^T e_j)\cap \hat{S}} (A_{ji} h^{(i)})^2 \,\Big|\, \eta_j = k \right] \\
&= \mathbf{E} \left[k \sum_{i\in\Omega(A^T e_j)\cap \hat{S}} (A_{ji} h^{(i)})^2 \,\Big|\, \eta_j = k \right] \\
&\overset{(58)}{=} \frac{k^2}{\omega} \sum_{i\in\Omega(A^T e_j)} (A_{ji} h^{(i)})^2. \quad (72)
\end{aligned}$$

Combining (71) and (72), we finally get

$$\begin{aligned}
\|\hat{A}\|^2_{w^*,\hat{v}} &\leq \frac{1}{\omega} \sum_{k=0}^{n} k^2 \pi_k \left(\max_{\|h\|_{w^*}=1} \sum_{j=1}^{m} v_j^{-2} \sum_{i=1}^{n} (A_{ji} h^{(i)})^2 \right) \\
&\overset{(\text{Lemma } 6)}{\leq} \frac{1}{\omega} \sum_{k=0}^{n} k^2 \pi_k \overset{(59)}{=} \frac{\tau}{n} \Big(1 + \frac{(\omega-1)(\tau-1)}{\max(1, n-1)} \Big).
\end{aligned}$$

(ii) Consider now the case $p = 1$.

$$\begin{aligned}
\|\hat{A}\|^2_{w^*,\hat{v}} &\overset{(69)+(12)+(42)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S^{-1}\left[\max_{1\le j\le m} v_j^{-2}\left(\sum_{i=1}^n \hat{A}^S_{ji}h^{(i)}\right)^2\right] \\
&\overset{(70)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S^{-1}\left[\max_{1\le j\le m} v_j^{-2}\left(\sum_{i=1}^n p_S\chi_{(i\in S)}A_{ji}h^{(i)}\right)^2\right] \\
&\overset{(19)}{=} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S\left[\max_{1\le j\le m} v_j^{-2}\Big(\sum_{i\in\Omega(A^Te_j)\cap S} A_{ji}h^{(i)}\Big)^2\right] \\
&\overset{\text{(Cauchy-Schwarz)}}{\le} \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S\left[\max_{1\le j\le m} v_j^{-2}|\Omega(A^Te_j)\cap S| \sum_{i\in\Omega(A^Te_j)\cap S} (A_{ji}h^{(i)})^2\right] \\
&\le \max_{\|h\|_{w^*}=1} \sum_{S\in\mathcal{P}} p_S\kappa_S\left[\max_{1\le j\le m} v_j^{-2}\sum_{i\in S}(A_{ji}h^{(i)})^2\right], \qquad (73)
\end{aligned}$$

where $\kappa_S \overset{\text{def}}{=} \max_{1\le j\le m}|\Omega(A^Te_j)\cap S|$. Consider the change of variables $y^{(i)} = (w_i^*)^{1/2}B_i^{1/2}h^{(i)}$. Utilizing essentially the same argument as in the proof of Lemma 6 for $p=1$, we obtain

$$\max_{1\le j\le m} v_j^{-2}\sum_{i\in S}(A_{ji}h^{(i)})^2 \le \sum_{i\in S}\|y^{(i)}\|_E^2. \qquad (74)$$

Since $\|y\|_E = \|h\|_{w^*}$, substituting (74) into (73) gives

$$\begin{aligned}
\|\hat{A}\|^2_{w^*,\hat{v}} &\le \max_{\|y\|_E=1}\sum_{S\in\mathcal{P}} p_S\kappa_S\sum_{i\in S}\|y^{(i)}\|_E^2 = \max_{\|y\|_E=1}\sum_{i=1}^n\|y^{(i)}\|_E^2\sum_{S\in\mathcal{P}} p_S\kappa_S\chi_{(i\in S)} \\
&= \max_{\|y\|_E=1}\sum_{i=1}^n\|y^{(i)}\|_E^2\mathbf{E}[\kappa_{\hat{S}}\chi_{(i\in\hat{S})}] \\
&= \max_{1\le i\le n}\mathbf{E}[\kappa_{\hat{S}}\chi_{(i\in\hat{S})}] = \max_{1\le i\le n}\mathbf{E}[\max_{1\le j\le m}|\Omega(A^Te_j)\cap\hat{S}|\times\chi_{(i\in\hat{S})}]. \qquad (75)
\end{aligned}$$

It now only remains to apply inequality (61) used with $J_j = \Omega(A^Te_j)$.

□

Let us now comment on some aspects of the above result.

1. It is possible to draw a link between β_2' and β_3'. In view of (59), for $p=2$ we have

$$\beta_2' = \tfrac{n}{\tau}\max_{1\le i\le n}\mathbf{E}[|\Omega(A^Te_j)\cap\hat{S}|\times\chi_{(i\in\hat{S})}],$$

where j is such that $|\Omega(A^Te_j)| = \omega$ (we can wlog assume this holds for all j). On the other hand, as is apparent from (75), for $p=1$ we can replace β_3' by

$$\beta_3'' \overset{\text{def}}{=} \tfrac{n}{\tau}\max_{1\le i\le n}\mathbf{E}[\max_{1\le j\le m}|\Omega(A^Te_j)\cap\hat{S}|\times\chi_{(i\in\hat{S})}].$$

Clearly, $\beta_2' \le \beta_3'' \le m\beta_2'$. However, in many situations, $\beta_3'' \approx \beta_2'$ (see Figure 1). Recall that a small β is good for Algorithm 1 (this will be formally proved in the next section).

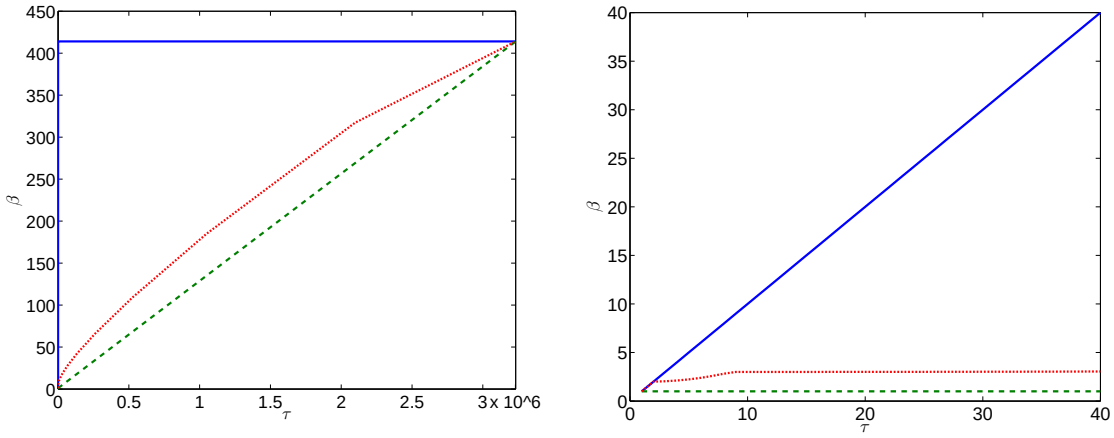

Figure 1: Comparison of the three formulae for β' as a function of the number of processors τ (smaller β' is better). We have used matrix $A \in \mathbb{R}^{m\times n}$ with $m = 2,396,130$, $n = 3,231,961$ and $\omega = 414$. **Blue solid line:** τ-uniform sampling, $\beta_1' = \min\{\omega, \tau\}$ (Theorem 13). **Green dashed line:** τ-nice sampling and $p = 2$, $\beta_2' = 1 + \frac{(\omega-1)(\tau-1)}{\max\{1,n-1\}}$ (Theorem 15). **Red dash-dotted line:** τ-nice sampling and $p = 1$, β_3' follows (68) in Theorem 15. Note that β_1' reaches its maximal value ω quickly, whereas β_2' increases slowly. When τ is small compared to n, this means that β_2' remains close to 1. As shown in Section 5 (see Theorems 17 and 16), small values of β' directly translate into better complexity and parallelization speedup. **Left:** Large number of processors. **Right:** Zoom for smaller number of processors.

2. If we let $\omega^* = \max_i\{j \ : \ A_{ji} \neq 0\}$ (maximum number of nonzero rows in matrices $A_1, \ldots, A_n$), then in the $p = 1$ case we can replace β_3' by the smaller quantity

$$\beta_3''' \stackrel{\text{def}}{=} \frac{\tau}{n} \sum_{k=k_{\min}}^{k_{\max}} \min\left\{1, \sum_{l=k}^{k_{\max}} \left(m\frac{n}{n-\omega}\frac{\tau-l}{\tau} + n\frac{\omega^*}{\omega}\frac{l}{\tau}\right)\pi_l\right\}.$$

5 Iteration Complexity

In this section we formulate *concrete* complexity results for Algorithm 1 applied to problem (7) by combining the generic results proved in [26] and outlined in the introduction, Lemma 3 (which draws a link between (7) and (8) and, *most importantly*, the concrete values of β and w established in this paper for Nesterov separable functions and τ-uniform and τ-nice samplings.

A function $\phi : \mathbb{R}^N \to \mathbb{R}\cup\{+\infty\}$ is strongly convex with respect to the norm $\|\cdot\|_w$ with convexity parameter $\sigma_\phi(w) \geq 0$ if for all $x, \bar{x} \in \operatorname{dom}\phi$,

$$\phi(x) \geq \phi(\bar{x}) + \langle \phi'(\bar{x}), x - \bar{x}\rangle + \frac{\sigma_\phi(w)}{2}\|x - \bar{x}\|_w^2,$$

where $\phi'(\bar{x})$ is any subgradient of ϕ at $\bar{x}$.

For $x_0 \in \mathbb{R}^N$ we let $\mathcal{L}_\mu^\delta(x_0) \stackrel{\text{def}}{=} \{x \ : \ F_\mu(x) \leq F_\mu(x_0) + \delta\}$ and let

$$\mathcal{D}_{w,\mu}^\delta(x_0) \stackrel{\text{def}}{=} \max_{x,y}\{\|x - y\|_w \ : \ x, y \in \mathcal{L}_\mu^\delta(x_0)\}$$

be the diameter of this set in the norm $\|\cdot\|_w$.

It will be useful to recall some basic notation from Section 2 that the theorems of this section will refer to: $F(x) = f(x) + \Psi(x)$, and $F_\mu(x) = f_\mu(x) + \Psi(x)$, with

$$f(x) = \max_{z\in Q}\{\langle Ax, z\rangle - g(z)\}, \quad f_\mu(x) = \max_{z\in Q}\{\langle Ax, z\rangle - g(z) - \mu d(z)\},$$

where d is a prox function on Q (it is strongly convex on Q wrt $\|\cdot\|_v$, with constant σ) and $D = \max_{x\in Q} d(x)$. Recall also that $\|\cdot\|_v$ is a weighted p norm on $\mathbb{R}^m$, with weights $v_1, \dots, v_m > 0$. Also recall that $\|\cdot\|_w$ is a norm defined as a weighted quadratic mean of the block-norms $\langle B_i x^{(i)}, x^{(i)}\rangle^{1/2}$, with weights $w_1, \dots, w_n > 0$.

Theorem 16 (Complexity: smoothed composite problem (8))**.** *Pick $x_0 \in \operatorname{dom}\Psi$ and let $\{x_k\}_{k\geq 0}$ be the sequence of random iterates produced by the smoothed parallel descent method (Algorithm 1) with the following setup:*

(i) $\{S_k\}_{k\geq 0}$ is an iid sequence of τ-uniform samplings, where $\tau \in \{1, 2, \dots, n\}$,

(ii) $w = w^*$, where w^* is defined in (38),

(iii) $\beta = \frac{\beta'}{\sigma\mu}$, where $\beta' = 1 + \frac{(\omega-1)(\tau-1)}{\max\{1,n-1\}}$ if the samplings are τ-nice and $p = 2$, β' is given by (68) if the samplings are τ-nice and $p = 1$, and $\beta' = \min\{\omega, \tau\}$ if the samplings are not τ-nice (ω is the degree of Nesterov separability).

Choose error tolerance $0 < \epsilon < F_\mu(x_0) - \min_x F_\mu(x)$, confidence level $0 < \rho < 1$ and iteration counter k as follows:

(i) if F_μ is strongly convex with $\sigma_{f_\mu}(w^*) + \sigma_\Psi(w^*) > 0$, choose

$$k \geq \frac{n}{\tau} \times \frac{\frac{\beta'}{\mu\sigma} + \sigma_\Psi(w^*)}{\sigma_{f_\mu}(w^*) + \sigma_\Psi(w^*)} \times \log\left(\frac{F_\mu(x_0) - \min_x F_\mu(x)}{\epsilon\rho}\right),$$

(ii) otherwise additionally assume[12] that $\epsilon < \frac{2n\beta}{\tau}$ and that[13] $\beta' = \min\{\omega, \tau\}$, and choose

$$k \geq \frac{n\beta'}{\tau} \times \frac{2(\mathcal{D}^0_{w^*,\mu}(x_0))^2}{\mu\sigma\epsilon} \times \log\left(\frac{F_\mu(x_0) - \min_x F_\mu(x)}{\epsilon\rho}\right).$$

Then

$$\mathbf{P}(F_\mu(x_k) - \min_x F_\mu(x) \leq \epsilon) \geq 1 - \rho.$$

Proof. This follows from the generic complexity bounds proved by Richtárik and Takáč [26, Theorem 19(ii) and Theorem 20] and Theorems 13 and 15 giving formulas for β' and w^* for which $(f_\mu, \hat{S}) \sim \text{ESO}(\frac{\beta'}{\sigma\mu}, w^*)$. □

[12] This assumption is not restrictive as $\beta' \geq 1$, $n \geq \tau$ and μ, σ are usually small. However, it is technically needed.

[13] Instead of the assumption $\beta' = \min\{\omega, \tau\}$ it suffices to include an additional step into SPCDM which accepts only updates decreasing the loss. That is, x_{k+1} is set x_k in case $F_\mu(x_{k+1}) > F_\mu(x_k)$. However, function evaluation is not recommended as it would considerable slow down the method. In our experiments we have never encountered a problem with using the more efficient τ-nice sampling even in the non-strongly convex case. In fact, this assumption may just be an artifact of the analysis.

We now we consider solving the nonsmooth problem (7) by applying Algorithm 1 to its smooth approximation (8) for a specific value of the smoothing parameter μ.

Theorem 17 (Complexity: nonsmooth composite problem (7))**.** *Pick $x_0 \in \operatorname{dom}\Psi$ and let $\{x_k\}_{k\geq 0}$ be the sequence of random iterates produced by the smoothed parallel descent method (Algorithm 1) with the same setup as in Theorem 16, where $\mu = \frac{\epsilon'}{2D}$ and $0 < \epsilon' < F(x_0) - \min_x F(x)$. Further, choose confidence level $0 < \rho < 1$ and iteration counter as follows:*

(i) if F_μ is strongly convex with $\sigma_{f_\mu}(w^*) + \sigma_\Psi(w^*) > 0$, choose

$$k \geq \frac{n}{\tau} \times \frac{\frac{2\beta' D}{\sigma\epsilon'} + \sigma_\Psi(w^*)}{\sigma_{f_\mu}(w^*) + \sigma_\Psi(w^*)} \times \log\left(\frac{2(F(x_0) - \min_x F(x)) + \epsilon'}{\epsilon'\rho}\right),$$

(ii) otherwise additionally assume that $(\epsilon')^2 < \frac{8nD\beta'}{\sigma\tau}$ and that $\beta' = \min\{\omega, \tau\}$, and choose

$$k \geq \frac{n\beta'}{\tau} \times \frac{8D(\mathcal{D}^{\epsilon'/2}_{w^*,0}(x_0))^2}{\sigma(\epsilon')^2} \times \log\left(\frac{2(F(x_0) - \min_x F(x)) + \epsilon'}{\epsilon'\rho}\right).$$

Then

$$\mathbf{P}(F(x_k) - \min_x F(x) \leq \epsilon') \geq 1 - \rho.$$

Proof. We will apply Theorem 16 with $\epsilon = \frac{\epsilon'}{2}$ and $\mu = \frac{\epsilon'}{2D}$. All that we need to argue in case (i) (and we need this in case (ii) as well) is: (a) $\epsilon < F_\mu(x_0) - F_\mu(x^*_\mu)$, where $x^*_\mu = \arg\min_x F_\mu(x)$ (this is needed to satisfy the assumption about ϵ), (b) $F_\mu(x_0) - F_\mu(x^*_\mu) \leq F(x_0) - F(x^*) + \epsilon$ (this is needed for logarithmic factor in the iteration counter) and (c) $\mathbf{P}(F_\mu(x_k) - F_\mu(x^*_\mu) \leq \epsilon) \leq \mathbf{P}(F(x_k) - F(x^*) \leq \epsilon')$, where $x^* = \arg\min_x F(x)$. Inequality (a) follows by combining our assumption with Lemma 3. Indeed, the assumption $\epsilon' < F(x_0) - F(x^*)$ can be written as $\frac{\epsilon'}{2} < F(x_0) - F(x^*) - \mu D$, which combined with the second inequality in (28), used with $x = x_0$, yields the result. Further, (b) is identical to the first inequality in Lemma 3 used with $x = x_0$. Finally, (c) holds since the second inequality of Lemma 3 with $x = x_k$ says that $F_\mu(x_k) - F_\mu(x^*_\mu) \leq \frac{\epsilon'}{2}$ implies $F(x_k) - F(x^*) \leq \epsilon'$.

In case (ii) we additionally need to argue that: (d) $\epsilon < \frac{2n\beta}{\tau}$ and (e) $\mathcal{D}^0_{w^*,\mu}(x_0) \leq \mathcal{D}^{\epsilon'/2}_{w^*,0}(x_0)$. Note that (d) is equivalent to the assumption $(\epsilon')^2 < \frac{8nD\beta'}{\sigma\tau}$. Notice that as long as $F_\mu(x) \leq F_\mu(x_0)$, we have

$$F(x) \overset{(24)}{\leq} F_\mu(x) + \tfrac{\epsilon'}{2} \leq F_\mu(x_0) + \tfrac{\epsilon'}{2} \overset{(24)}{\leq} F(x_0) + \tfrac{\epsilon'}{2},$$

and hence $\mathcal{L}^0_\mu(x_0) \subset \mathcal{L}^{\epsilon'/2}_0(x_0)$, which implies (e). □

Let us now briefly comment on the results.

1. If we choose the separable regularizer $\Psi(x) = \frac{\delta}{2}\|x\|^2_{w^*}$, then $\sigma_\Psi(w^*) = \delta$ and the strong convexity assumption is satisfied, irrespective of whether f_μ is strongly convex or not. A regularizer of this type is often chosen in machine learning applications.

2. Theorem 17 covers the problem $\min F(x)$ and hence we have on purpose formulated the results without any reference to the smoothed problem (with the exception of dependence on $\sigma_{f_\mu}(w^*)$ in case (i)). We traded a (very) minor loss in the quality of the results for a more direct formulation.

3. As the confidence level is inside a logarithm, it is easy to obtain a high probability result with this randomized algorithm. For problem (7) in the non-strongly convex case, iteration complexity is $O((\epsilon')^{-2})$ (ignoring the logarithmic term), which is comparable to other techniques available for the minimization of nonsmooth convex functions such as the subgradient method. In the strongly convex case the dependence is $O((\epsilon')^{-1})$. Note, however, that in many applications solutions only of moderate or low accuracy are required, and the focus is on the dependence on the number of processors τ instead. In this regard, our methods have excellent theoretical parallelization speedup properties.

4. It is clear from the complexity results that as more processors τ are used, the method requires fewer iterations, and the speedup gets higher for smaller values of ω (the degree of Nesterov separability of f). However, the situation is even better if the regularized Ψ is strongly convex – the degree of Nesterov separability then has a weaker effect on slowing down parallelization speedup.

5. For τ-nice samplings, β changes depending on p (the type of dual norm $\|\cdot\|_v$). However, σ changes also, as this is the strong convexity constant of the prox function d with respect to the dual norm $\|\cdot\|_v$.

6 Computational Experiments

In this section we consider the application of the smoothed parallel coordinate descent method (SPCDM) to three special problems and comment on some *preliminary* computational experiments. For simplicity, in all examples we assume all blocks are of size 1 ($N_i = 1$ for all i) and fix $\Psi \equiv 0$.

In all tests we used a shared-memory workstation with 32 Intel Xeon processors at 2.6 GHz and 128 GB RAM. We coded an asynchronous version of SPCDM to limit communication costs and approximated τ-nice sampling by a τ-independent sampling as in [26] (the latter is very easy to generate in parallel).

6.1 L-infinity regression / linear programming

Here we consider the the problem of minimizing the function

$$f(x) = \|\tilde{A}x - \tilde{b}\|_\infty = \max_{u \in Q} \{\langle Ax, u\rangle - \langle b, u\rangle\},$$

where

$$\tilde{A} \in \mathbb{R}^{m\times n}, \qquad \tilde{b} \in \mathbb{R}^m, \qquad A = \begin{bmatrix} \tilde{A} \\ -\tilde{A} \end{bmatrix} \in \mathbb{R}^{2m\times n}, \qquad b = \begin{bmatrix} \tilde{b} \\ -\tilde{b} \end{bmatrix} \in \mathbb{R}^{2m}$$

and $Q \overset{\text{def}}{=} \{u_j \in \mathbb{R}^{2m} \ : \ \sum_j u_j = 1,\ u_j \geq 0\}$ is the unit simplex in $\mathbb{R}^{2m}$. We choose the dual norm $\|\cdot\|_v$ in $\mathbb{R}^{2m}$ with $p = 1$ and $v_j = 1$ for all j. Further, we choose the prox function $d(u) = \log(2m) + \sum_{j=1}^{2m} u_j \log(u_j)$ with center $u_0 = (1, 1, \ldots, 1)/(2m)$. It can be shown that $\sigma = 1$ and $D = \log(2m)$. Moreover, we let all blocks be of size 1 ($N_i = 1$), choose $B_i = 1$ for all i in the definition of the primal norm and

$$w_i^* \overset{(29)}{=} \max_{1\leq j\leq 2m} A_{ji}^2 = \max_{1\leq j\leq m} \tilde{A}_{ji}^2 \ .$$

The smooth approximation of f is given by

$$f_\mu(x) = \mu \log\left(\frac{1}{2m}\sum_{j=1}^{2m}\exp\left(\frac{e_j^T Ax - b_j}{\mu}\right)\right) \quad . \tag{76}$$

Experiment. In this experiment we minimize f_μ utilizing τ-nice sampling and parameter β given by (68). We first compare SPCDM (Algorithm 1) with several other methods, see Table 1.

We perform a small scale experiment so that we can solve the problem directly as a linear program with GLPK. The simplex method struggles to progress initially but eventually finds the exact solution quickly. The accelerated gradient algorithm of Nesterov is easily parallelizable, which makes it competitive, but it suffers from small stepsizes (we chose here the estimate for the Lipschitz constant of the gradient given in [20] for this problem). A very efficient algorithm for the minimization of the infinity norm is Nesterov's sparse subgradient method [20] that is the fastest in our tests even when it uses a single core only. It performs full subgradient iterations in a very cheap way, utilizing the fact that the subgradients are sparse. The method has a sublinear in n complexity. However, in order for the method to take long steps, one needs to know the optimal value in advance. Otherwise, the algorithm is much slower, as is shown in the table.

Algorithm	*# iterations*	*time (second)*
GLPK's simplex	55,899	681
Accelerated gradient [18], $\tau = 16$ cores	8,563	246
Sparse subgradient [20], optimal value known	1,730	6.4
Sparse subgradient [20], optimal value unknown	166,686	544
Smoothed PCDM (Theorem 16), $\tau = 4$ cores ($\beta = 3.0$)	15,700,000	53
Smoothed PCDM (Theorem 16), $\tau = 16$ cores ($\beta = 5.4$)	7,000,000	37

Table 1: Comparison of various algorithms for the minimization of $f(x) = \|\tilde{A}x - \tilde{b}\|_\infty$, where $\tilde{A}$ and $\tilde{b}$ are taken from the Dorothea dataset [7] ($m = 800$, $n = 100,000$, $\omega = 6,061$) and $\epsilon = 0.01$.

For this problem, and without the knowledge of the optimal value, the smoothed parallel coordinate descent method presented in this paper is the fastest algorithm. Many iterations are needed but they are very cheap: in its serial version, at each iteration one only needs to compute one partial derivative and to update 1 coordinate of the optimization variable, the residuals and the normalization factor. The worst case algorithmic complexity of one iteration is thus proportional to the number of nonzero elements in one column; on average.

Observe that quadrupling the number of cores does not divide by 4 the computational time because of the increase in the β parameter. Also note that we have tested our method using the parameters dictated by the theory. In our experience the performance of the method imroves for a smaller value of β: this leads to larger stepsizes and the method often tolerates this.

Remark: There are numerical issues with the smooth approximation of the infinity norm because it involves exponentials of potentially large numbers. A safe way of computing (76) is to compute first $\bar{r} = \max_{1\le j\le 2m}(Ax-b)_j$ and to use the safe formula

$$f_\mu(x) = \bar{r} + \mu \log\left(\frac{1}{2m}\sum_{j=1}^{2m}\exp\left(\frac{(Ax-b)_j - \bar{r}}{\mu}\right)\right).$$

However, this formula is not suitable for parallel updates because the logarithm prevents us from making reductions. We adapted it in the following way to deal with parallel updates. Suppose we have already computed $f_\mu(x)$. Then

$f_\mu(x+h) = f_\mu(x) + \mu \log(S_x(\delta))$, where

$$S_x(h) \stackrel{\text{def}}{=} \frac{1}{2m} \sum_{j=1}^{2m} \exp\left(\frac{(Ax-b)_j + (Ah)_j - f_\mu(x)}{\mu} \right)$$

In particular, $S_x(0) = 1$. Thus, as long as the updates are reasonably small, one can compute $\exp[((Ax-b)_j + (Ah)_j - f_\mu(x))/\mu]$ and update the sum in parallel. From time to time (for instance every n iterations or when S_x becomes small), we recompute $f_\mu(x)$ from scratch and reset h to zero.

6.2 L1 regression

Here we consider the problem of minimizing the function

$$f(x) = \|Ax-b\|_1 = \max_{u \in Q} \{\langle Ax, u\rangle - \langle b, u\rangle\},$$

where $Q = [-1,1]^n$. We define the dual norm $\|\cdot\|_v$ with $p=2$ and $v_j = \sum_{i=1}^n A_{ji}^2$ for all $j = 1,2,\dots,m$. Further, we choose the prox function $d(z) = \frac{1}{2}\|z\|_v^2$ with center $z_0 = 0$. Clearly, $\sigma = 1$ and $D = \frac{1}{2}\sum_{j=1}^m v_j = \sum_{j=1}^m \sum_{i=1}^n A_{ji}^2 = \|A\|_F^2$. Moreover, we choose $B_i = 1$ for all $i = 1,2,\dots,n$ in the definition of the primal norm and

$$w_i^* \stackrel{(29)}{=} \sum_{j=1}^m v_j^{-2} A_{ji}^2, \qquad i = 1,2,\dots,n.$$

The smooth approximation of f is given by

$$f_\mu(x) = \sum_{j=1}^m \|e_j^T A\|_{w^*}^* \psi_\mu\left(\frac{|e_j^T Ax - b_j|}{\|e_j^T A\|_v^*} \right), \qquad \psi_\mu(t) = \begin{cases} \frac{t^2}{2\mu}, & 0 \le t \le \mu, \\ t - \frac{\mu}{2}, & \mu \le t. \end{cases}$$

Remark: Note that in [18], the dual norm is defined from the primal norm. In the present work, we need to define the dual norm first since otherwise the definitions of the norms would cycle. However, the definitions above give the choice of v that minimizes the term

$$D\|e\|_{w^*}^2 = D\sum_{i=1}^n w_i^* = (\sum_{j=1}^m v_j)(\sum_{j'=1}^m v_{j'}^{-2} A_{j'i}^2),$$

where $e = (1,1,\dots,1) \in \mathbb{R}^N$. We believe that in the non-strongly convex case one can replace in the complexity estimates the squared diameter of the level set by $\|x_0 - x^*\|_{w^*}^2$, which would then mean that a product of the form $D\|x_0 - x_*\|_{w^*}^2$ appears in the complexity. The above choice of the weights $v_1,\dots,v_m$ minimizes this product under assuming that $x_0 - x^*$ is proportional to e.

Experiment. We performed our medium scale numerical experiments (in the case of L1 regression and exponential loss minimization (Section 6.3)) on the URL reputation dataset [12]. It gathers $n = 3,231,961$ features about $m = 4,792,260$ URLs collected during 120 days. The feature matrix is sparse but it has some dense columns. The maximum number of nonzero elements in a row is $\omega = 414$. The vector of labels classifies the page as spam or not.

We applied SPCDM with τ-nice sampling, following the setup described in Theorem 17. The results for $f(x) = \|Ax-b\|_1$ are gathered in Figure 2. We can see that parallelization speedup is proportional to the number of processors. In the right plot we observe that the algorithm is not monotonic but monotonic on average.

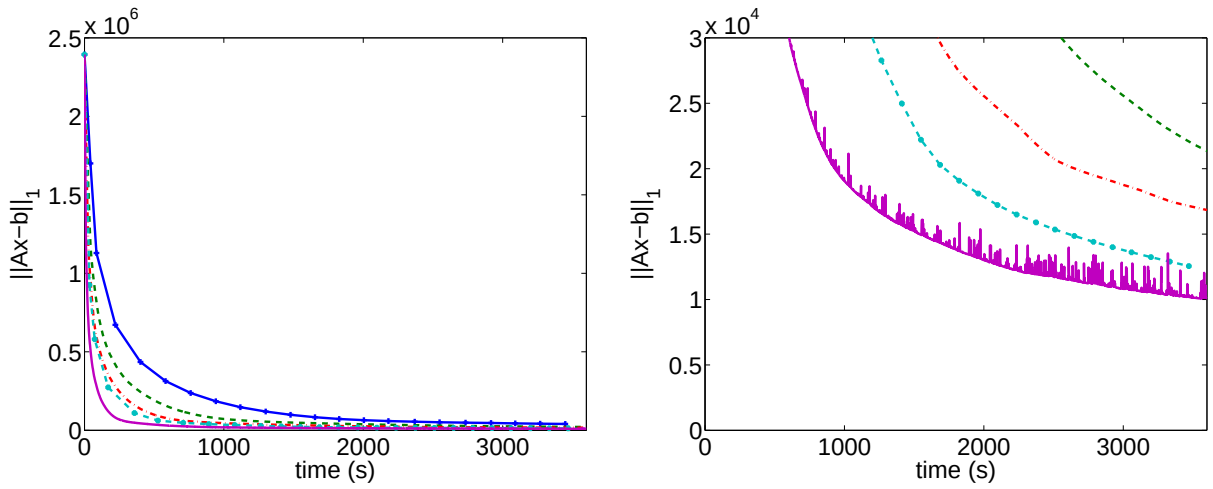

Figure 2: Performance of SPCDM on the problem of minimizing $f(x) = \|Ax - b\|_1$ where A and b are given by the URL reputation dataset. We have run the method until the function value was decreased by a factor of 240. **blue solid line with crosses:** $\tau = 1$; **green dashed line:** $\tau = 2$; **red dash-dotted line:** $\tau = 4$; **cyan dashed line with stars:** $\tau = 8$; **solid purple line:** $\tau = 16$. **Left:** Decrease of the objective value in time. We can see that parallelization speedup is proportional to the number of processors. **Right:** Zoom on smaller objective values. We can see that the algorithm is not monotonic but monotonic on average.

6.3 Logarithm of the exponential loss

Here we consider the problem of minimizing the function

$$f_1(x) = \log\left(\frac{1}{m}\sum_{j=1}^{m}\exp(b_j(Ax)_j)\right). \tag{77}$$

The AdaBoost algorithm [6] minimizes the exponential loss $\exp(f_1(x))$ by a greedy serial coordinate descent method (i.e., at each iteration, one selects the coordinate corresponding to the largest directional derivative and updates that coordinate only). Here we observe that f_1 is Nesterov separable as it is the smooth approximation of

$$f(x) = \max_{1\le j\le m} b_j(Ax)_j$$

with $\mu = 1$. Hence, we can minimize f_1 by parallel coordinate descent with τ-nice sampling and β given by (67).

Convergence of AdaBoost is not a trivial result because the minimizing sequences may be unbounded. The proof relies on a decomposition of the optimization variables to an unbounded part and a bounded part [14, 37]. The original result gives iteration complexity $O(\frac{1}{\epsilon})$.

Parallel versions of AdaBoost have previously been studied. I our notation, Collins, Shapire and Singer [3] use $\tau = n$ and $\beta = \omega$. Palit and Reddy [22] use a generalized greedy sampling and take $\beta = \tau$ (number of processors). In the present work, we use randomized samplings and we can take $\beta \ll \min\{\omega, \tau\}$ with the τ-nice sampling. As discussed before, this value of β can be $O(\sqrt{n})$

times smaller than $\min\{\omega, \tau\}$, which leads to big gains in iteration complexity. For a detailed study of the properties of the SPCDM method applied to the AdaBoost problem we refer to a follow up work of Fercoq [5].

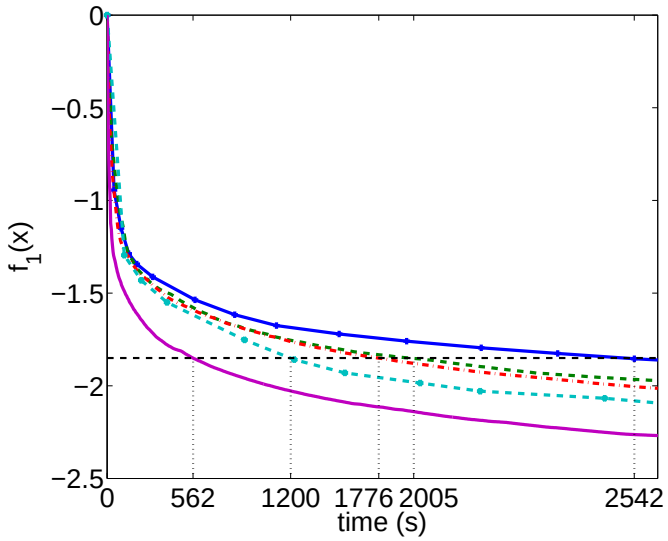

Figure 3: Performance of the smoothed parallel coordinate descent method (SPCDM) with $\tau = 1, 2, 4, 8, 16$ processors, applied to the problem of minimizing the logarithm of the exponential loss (77), where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ are given by the URL reputation dataset; $m = 7,792,260$, $n = 3,231,961$ and $\omega = 414$. When $\tau = 16$ processors were used, the method needed 562s to obtain a solution of a given accuracy (depicted by the horizontal line). When $\tau = 8$ processors were used, the method needed 1200s, roughly double that time. Compared to a single processor, which needed 2542s, the setup with $\tau = 16$ was nearly 5 times faster. Hence, it is possible to observe nearly parallelization speedup, as our theory predicts. Same colors were used as in Figure 2.

Experiment. In our last experiment we demonstrate how SPCDM (which can be viewed as a random parallel version of AdaBoost) performs on the URL reputation dataset. Looking at Figure 3, we see that parallelization leads to acceleration, and the time needed to decrease the loss to -1.85 is inversely proportional to the number of processors. Note that the additional effort done by increasing the number of processors from 4 to 8 is compensated by the increase of β from 1.2 to 2.0 (this is the little step in the zoom of Figure 1). Even so, further acceleration takes place when one further increases the number of processors.

References

[1] Yatao Bian, Xiong Li, and Yuncai Liu. Parallel coordinate descent newton for large-scale l1-regularized minimization. *arXiv1306:4080v1*, June 2013.

[2] Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for L1-regularized loss minimization. In *28th International Conference on Machine Learning*, 2011.

[3] Michael Collins, Robert E. Shapire, and Yoram Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

[4] Cong D. Dang and Lan Guanghui. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. Technical report, Georgia Institute of Technology, September 2013.

[5] Olivier Fercoq. Parallel coordinate descent for the AdaBoost problem. In *International Conference on Machine Learning and Applications - ICMLA'13*, 2013.

[6] Yoav Freund and Robert E. Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pages 23–37. Springer, 1995.

[7] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. *Advances in Neural Information Processing Systems*, 17:545–552, 2004.

[8] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

[9] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletcher. Block-coordinate frank-wolfe optimization for structural svms. In *30th International Conference on Machine Learning*, 2013.

[10] Dennis Leventhal and Adrian S. Lewis. Randomized methods for linear constraints: Convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[11] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. Technical report, Microsoft Research, 2013.

[12] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 681–688. ACM, 2009.

[13] Indraneel Mukherjee, Kevin Canini, Rafael Frongillo, and Yoram Singer. Parallel boosting with momentum. Technical report, Google Inc., 2013.

[14] Indraneel Mukherjee, Cynthia Rudin, and Robert E. Shapire. The rate of convergence of AdaBoost. *arXiv:1106.6024*, 2011.

[15] Ion Necoara and Dragos Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed mpc. *Journal of Process Control*, 23:243–253, 2013.

[16] Ion Necoara, Yurii Nesterov, and Francois Glineur. Efficiency of randomized coordinate descent methods on optimization problems with linearly coupled constraints. Technical report, Politehnica University of Bucharest, 2012.

[17] Ion Necoara and Andrei Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Technical report, University Politehnica Bucharest, 2012.

[18] Yurii Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

[19] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[20] Yurii Nesterov. Subgradient methods for huge-scale optimization problems. *CORE DISCUSSION PAPER 2012/2*, 2012.

[21] Yurii Nesterov. Gradient methods for minimizing composite function. *Mathematical Programming*, 140(1):125–161, 2013.

[22] Indranil Palit and Chandan K. Reddy. Scalable and parallel boosting with MapReduce. *IEEE Transactions on Knowledge and Data Engineering*, 24(10):1904–1916, 2012.

[23] Peter Richtárik and Martin Takáč. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In *Operations Research Proceedings*, pages 27–32. Springer, 2012.

[24] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2012.

[25] Peter Richtárik and Martin Takáč. Efficiency of randomized coordinate descent methods on minimization problems with a composite objective function. In *4th Workshop on Signal Processing with Adaptive Sparse Structured Representations*, June 2011.

[26] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization problems. *arXiv:1212.0873*, November 2012.

[27] Peter Richtárik, Martin Takáč, and S. Damla Ahipaşaoğlu. Alternating maximization: unifying framework for 8 sparse PCA formulations and efficient parallel codes. *arXiv:1212:4137*, December 2012.

[28] Andrzej Ruszczyński. On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Reseach*, 20(3):634–656, 1995.

[29] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[30] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

[31] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. *arXiv:1305.2581v1*, May 2013.

[32] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

[33] Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, 2013.

[34] Qing Tao, Kang Kong, Dejun Chu, and Gaowei Wu. Stochastic coordinate descent methods for regularized smooth and nonsmooth losses. *Machine Learning and Knowledge Discovery in Databases*, pages 537–552, 2012.

[35] Rachael Tappenden, Peter Richtárik, and Burak Büke. Separable approximations and decomposition methods for the augmented Lagrangian. *arXiv:1308.6774*, August 2013.

[36] Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. *arXiv:1304.5530*, April 2013.

[37] Matus Telgarsky. A primal-dual convergence analysis of boosting. *The Journal of Machine Learning Research*, 13:561–606, 2012.